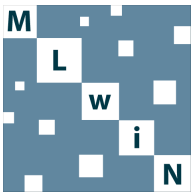
A User's Guide to MLwiN

Version 2.32



by Jon Rasbash, Fiona Steele,

William J. Browne & Harvey Goldstein

Centre for Multilevel Modelling,
University of Bristol

Programming by Jon Rasbash,
Chris Charlton & William J. Browne

A User's Guide to MLwiN

Copyright © 2015 Jon Rasbash, Fiona Steele, William J. Browne and Harvey Goldstein. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, for any purpose other than the owner's personal use, without the prior written permission of one of the copyright holders.

ISBN: 978-0-903024-97-6

Printed in the United Kingdom

First Printing November 2004.

Updated for University of Bristol, October 2005, February 2009, September 2012, June 2014, August 2014 and January 2015.

This manual is dedicated to the memory of Ian Langford, a greatly missed friend and colleague.

0.1 A Brief Orientation

MLwiN comes with the following main Manuals:

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2014) *A User's Guide to MLwiN, v2.31.* Centre for Multilevel Modelling, University of Bristol.

Rasbash, J., Charlton, C., Jones, K. and Pillinger, R. (2014) *Manual Supplement to MLwiN v2.31.* Centre for Multilevel Modelling, University of Bristol.

Browne, W.J. (2014) *MCMC Estimation in MLwiN, v2.31.* Centre for Multilevel Modelling, University of Bristol.

If you are new to multilevel modelling we recommend that you begin by following the Multilevel modelling online course (`http://www.bristol.ac.uk/cmm/learning/online-course/index.html`) especially Module 5 or use this User Manual, Chapter 1

If you are new to MLwiN we recommend that you first work your way through the User's Guide and cover Chapter 1 through 8; this will give you a good systematic understanding of the model for hierarchical structures estimated by the IGLS (likelihood) algorithm.

If you want to fit discrete outcome models, then first follow the online training materials (beginning with Module 6), then the relevant chapters of the User's Guide, and then corresponding chapters in the MCMC Manual. In a number of circumstances the MCMC estimation procedures will produce less-biased estimates than those produced by IGLS using quasi-likelihood.

If you want to analyse non-hierarchical structures, then we recommend that you start with the online training materials (modules 12 and 13) and then use the MCMC Manual as this form of estimation is needed for models with complex (non-hierarchical) structure. The MCMC Manual is also the starting place for multilevel factor analysis, models in which a continuous predictor is measured with error, and spatial models. The earlier chapters of this Manual act as a thorough introduction to the use of MCMC estimation. It is also important to consider Chapters 21 to 25 which examine how to speed up estimation through less-correlated chains.

If you want to impute for models with missing data, we suggest that you look at the Stat-JR software at
`http://www.bristol.ac.uk/cmm/software/statjr/`.

We are in the process of updating our manuals and training materials but inevitably there is some lag as there is now a very large amount of material.

Consequently we have produced a guide to the best place to look for help and assistance which we will continue to update on a regular basis:

`http://www.bristol.ac.uk/cmm/software/mlwin/mlwin-resources.html`

This website points you where to look for further detail and gives an overview of all our materials. Where there are multiple entries for a topic, we have tried to put the most introductory first, followed by the most comprehensive; the most technical comes at the end.

0.2 The purpose of this User Guide

The purpose of this Manual is to facilitate your introduction to the fitting of multilevel models in MLwiN. Chapters 1 to 7 form the basic core of the material and provide a 'hands-on approach' illustrated with data (worksheets) that are supplied with the program. Chapter 8 discusses how to use your own data. The following chapters are aimed to be relatively self-contained after the core material has been covered. Thus Chapters 9 to 12 deal with the analysis of models for discrete outcomes (eg for Yes/No responses and counts), Chapter 13 on repeated measures panel data and Chapter 14 on multivariate models with more than one response. Chapter 15 deals with diagnostic procedures for examining model assumptions.

Up to this point estimation has been through likelihood (IGLS), restricted (RIGLS) and quasi-likelihood procedures (for discrete outcome models) and for strictly hierarchical models. In contrast, Chapter 16 gives a short introduction to models estimated by MCMC procedures which are the subject of the complete MCMC Manual which is the more definitive source. Chapter 17 covers bootstrap estimation while Chapters 18 and 19 introduce different forms of non-hierarchical models as estimated by likelihood procedures. A much fuller treatment of these non-hierarchical models is given in the MCMC manual and that is how these more complex types of models are now generally estimated.

0.3 The purpose of the Manual Supplement

The Manual Supplement has been written to document and illustrate new features that were added in Version 2 of MLwiN.

We would like to draw your attention particularly to five facilities that will be of general use across all models:

- Section 1: improved facilities for specifying the right- hand side of the

model this allows for coding of categorical predictors, interactions and polynomials;

- Section 2: the Customised Predictions facility is used post-estimation so that you can readily characterize the results of a model in a table and/or graphically; this is particularly useful for models with interactions and for discrete outcome models (allowing both cluster specific and population average predictions), and for multivariate models with more than one response variable;
- Section 4: the Model Comparison facility allows you to compare a sequence of models; we have also added particular capabilities for summarizing models estimated by MCMC;
- Section 8 details the Names window which allows you control over columns in the worksheet;

0.4 Commands and macros

It is possible to use MLwiN via commands and the **Command Manual** provides a guide (Rasbash, J., Browne, W.J. and Goldstein, H. (2003) *MLwiN 2.0 Command Manual*. University of Bristol). More recently added commands are covered in the **Manual Supplement**. Finally it is also possible to call MLwiN from other software:

- **R2MLwiN** is an R command allowing users to fit multilevel models using MLwiN from within the R environment;
- **Stat-JR** is a software environment for promoting interactive complex statistical modelling; it has its own estimation MCMC procedures but can also call MLwiN;
- **RunMLwiN** is a Stata command to fit multilevel models in MLwiN from within Stata.

Up to date information is available from the CMM website.

Contents

Table of Contents **xii**

Introduction **xiii**
About the Centre for Multilevel Modelling . . . xiii
Installing the MLwiN software . . . xiii
MLwiN overview . . . xiv
Enhancements in Version 2.26 . . . xv
Estimation . . . xv
Exploring, importing and exporting data . . . xv
Improved ease of use . . . xvi
MLwiN Help . . . xvi
Compatibility with existing MLn software . . . xvi
Macros . . . xvii
The structure of the User's Guide . . . xvii
Acknowledgements . . . xvii
Further information about multilevel modelling . . . xviii
Technical Support . . . xviii

1 Introducing Multilevel Models **1**
1.1 Multilevel data structures . . . 1
1.2 Consequences of ignoring a multilevel structure . . . 2
1.3 Levels of a data structure . . . 3
1.4 An introductory description of multilevel modelling . . . 6

2 Introduction to Multilevel Modelling **9**
2.1 The tutorial data set . . . 9
2.2 Opening the worksheet and looking at the data . . . 10
2.3 Comparing two groups . . . 13
2.4 Comparing more than two groups: Fixed effects models . . . 20
2.5 Comparing means: Random effects or multilevel model . . . 28
Chapter learning outcomes . . . 35

3 Residuals **37**
3.1 What are multilevel residuals? . . . 37
3.2 Calculating residuals in MLwiN . . . 40
3.3 Normal plots . . . 43
Chapter learning outcomes . . . 45

4 Random Intercept and Random Slope Models **47**

4.1 Random intercept models . . . 47
4.2 Graphing predicted school lines from a random intercept model 51
4.3 The effect of clustering on the standard errors of coefficients . 58
4.4 Does the coefficient of **standlrt** vary across schools? Introducing a random slope . . . 59
4.5 Graphing predicted school lines from a random slope model . 63
Chapter learning outcomes . . . 64

5 Graphical Procedures for Exploring the Model 67
5.1 Displaying multiple graphs . . . 67
5.2 Highlighting in graphs . . . 70
Chapter learning outcomes . . . 79

6 Contextual Effects 81
6.1 The impact of school gender on girls' achievement . . . 82
6.2 Contextual effects of school intake ability averages . . . 85
Chapter learning outcomes . . . 89

7 Modelling the Variance as a Function of Explanatory Variables 91
7.1 A level 1 variance function for two groups . . . 91
7.2 Variance functions at level 2 . . . 97
7.3 Further elaborating the model for the student-level variance . 101
Chapter learning outcomes . . . 108

8 Getting Started with your Data 109
8.1 Inputting your data set into MLwiN . . . 109
Reading in an ASCII text data file . . . 109
Common problems that can occur in reading ASCII data from a text file . . . 110
Pasting data into a worksheet from the clipboard . . . 111
Naming columns . . . 112
Adding category names . . . 113
Missing data . . . 113
Unit identification columns . . . 114
Saving the worksheet . . . 114
Sorting your data set . . . 114
8.2 Fitting models in MLwiN . . . 117
What are you trying to model? . . . 117
Do you really need to fit a multilevel model? . . . 117
Have you built up your model from a variance components model? . . . 118
Have you centred your predictor variables? . . . 118
Chapter learning outcomes . . . 118

9 Logistic Models for Binary and Binomial Responses 119
9.1 Introduction and description of the example data . . . 119
9.2 Single-level logistic regression . . . 121
Link functions . . . 121

Interpretation of coefficients . . . 122
Fitting a single-level logit model in MLwiN . . . 122
A probit model . . . 128
9.3 A two-level random intercept model . . . 129
Model specification . . . 129
Estimation procedures . . . 130
Fitting a two-level random intercept model in MLwiN . . . 130
Variance partition coefficient . . . 133
Adding further explanatory variables . . . 136
9.4 A two-level random coefficient model . . . 137
9.5 Modelling binomial data . . . 141
Modelling district-level variation with district-level proportions 141
Creating a district-level data set . . . 142
Fitting the model . . . 144
Chapter learning outcomes . . . 145

10 Multinomial Logistic Models for Unordered Categorical Responses 147
10.1 Introduction . . . 147
10.2 Single-level multinomial logistic regression . . . 148
10.3 Fitting a single-level multinomial logistic model in MLwiN . . 149
10.4 A two-level random intercept multinomial logistic regression model . . . 156
10.5 Fitting a two-level random intercept model . . . 157
Chapter learning outcomes . . . 161

11 Fitting an Ordered Category Response Model 163
11.1 Introduction . . . 163
11.2 An analysis using the traditional approach . . . 164
11.3 A single-level model with an ordered categorical response variable . . . 168
11.4 A two-level model . . . 173
Chapter learning outcomes . . . 183

12 Modelling Count Data 185
12.1 Introduction . . . 185
12.2 Fitting a simple Poisson model . . . 186
12.3 A three-level analysis . . . 188
12.4 A two-level model using separate country terms . . . 190
12.5 Some issues and problems for discrete response models . . . 194
Chapter learning outcomes . . . 194

13 Fitting Models to Repeated Measures Data 195
13.1 Introduction . . . 195
13.2 A basic model . . . 198
13.3 A linear growth curve model . . . 205
13.4 Complex level 1 variation . . . 208
13.5 Repeated measures modelling of non-linear polynomial growth 208
Chapter learning outcomes . . . 212

14 Multivariate Response Models 213
14.1 Introduction . . . 213
14.2 Specifying a multivariate model . . . 214
14.3 Setting up the basic model . . . 216
14.4 A more elaborate model . . . 221
14.5 Multivariate models for discrete responses . . . 224
Chapter learning outcomes . . . 226

15 Diagnostics for Multilevel Models 229
15.1 Introduction . . . 229
15.2 Diagnostics plotting: Deletion residuals, influence and leverage 235
15.3 A general approach to data exploration . . . 244
Chapter learning outcomes . . . 244

16 An Introduction to Simulation Methods of Estimation 245
16.1 An illustration of parameter estimation with Normally distributed data . . . 246
16.2 Generating random numbers in MLwiN . . . 253
Chapter learning outcomes . . . 257

17 Bootstrap Estimation 259
17.1 Introduction . . . 259
17.2 Understanding the iterated bootstrap . . . 260
17.3 An example of bootstrapping using MLwiN . . . 261
17.4 Diagnostics and confidence intervals . . . 268
17.5 Nonparametric bootstrapping . . . 268
Chapter learning outcomes . . . 274

18 Modelling Cross-classified Data 275
18.1 An introduction to cross-classification . . . 275
18.2 How cross-classified models are implemented in MLwiN . . . 277
18.3 Some computational considerations . . . 277
18.4 Modelling a two-way classification: An example . . . 279
18.5 Other aspects of the SETX command . . . 281
18.6 Reducing storage overhead by grouping . . . 283
18.7 Modelling a multi-way cross-classification . . . 284
18.8 MLwiN commands for cross-classifications . . . 285
Chapter learning outcomes . . . 286

19 Multiple Membership Models 287
19.1 A simple multiple membership model . . . 287
19.2 MLwiN commands for multiple membership models . . . 290
Chapter learning outcomes . . . 290

Bibliography 291

Index 294

Introduction

About the Centre for Multilevel Modelling

The Centre for Multilevel Modelling was established in 1986, and has been supported largely by project grants from the UK Economic and Social Research Council. The Centre has been based at the University of Bristol since 2005. Members of the Bristol team can be found on this page:

`http://www.bristol.ac.uk/cmm/team/`

Centre contact details:

Centre for Multilevel Modelling
Graduate School of Education
University of Bristol
2 Priory Road
Bristol
BS8 1TX
United Kingdom

e-mail: info-cmm@bristol.ac.uk

T/F: +44(0)117 3310833

Installing the MLwiN software

MLwiN will install under Windows XP, Vista, 7 or 8. The installation procedure is as follows.

Run the file **MLwiN.msi** from wherever you have downloaded it to, or from the CD you have been sent. You will be guided through the installation procedure. Once installed you simply run MLwiN.exe, or for example, create a shortcut menu item for it on your desktop.

MLwiN overview

MLwiN is a development from MLn and its precursor, ML3, which provided a system for the specification and analysis of a range of multilevel models. MLwiN provides a graphical user interface (GUI) for specifying and fitting a wide range of multilevel models, together with plotting, diagnostic and data manipulation facilities. The user can carry out tasks by directly manipulating GUI screen objects, for example, equations, tables and graphs.

The computing module of MLwiN is effectively a somewhat modified version of the DOS MLn program, which is driven by a series of commands and operates in the background. Users typically will set about their modelling tasks by directly manipulating the GUI screen objects. The GUI translates these user actions into MLn commands, which are then sent to the computing module. When the computing module has completed the requested action all relevant GUI windows are notified of this and redraw themselves to reflect the updated system state. For some more complex models and tasks, for which there are currently no GUI structures available, the user must enter commands directly in the **Command interface** window. Any commands issued by the GUI are also recorded in this window. All these commands are fully described in the MLwiN **Help** system (see below).

It is assumed that you have a working knowledge of Windows applications. The MLwiN interface shares many features common to other applications such as word processors and some statistical packages. Thus, file opening and saving is standard, as is the arranging and copying of windows to the clipboard, and using menus and dialogue boxes.

The data structure is essentially that of a spreadsheet with columns denoting variables and rows corresponding to the lowest level units in the hierarchy. For example in the data set described in Chapter 2, there are 4059 rows, one for each student, and there are columns identifying students and schools and containing the values of the variables used in the analysis. By default the program allocates 1500 columns, 150 fixed and 150 random parameters and 5 levels of nesting. The worksheet dimensions, the number of parameters and the number of levels can be allocated dynamically.

For your own data analysis, typically you will have prepared your data in rows (or records) corresponding to the cases you have observed. MLwiN enables such data to be read into separate columns of a new worksheet, one column for each field. Data input and output is accessed from the **File** menu.

Other columns may be used for other purposes, for example to hold frequency data or parameter estimates, and need not be of the same length. Columns are numbered and can be referred to either as c1, c17, c43 etc., or by name if they have previously been named using the NAME feature in the **Names** window. MLwiN also has groups whose elements are a set of columns. These

are fully described in the MLwiN **Help** system.

As well as the columns there are also *boxes* or constants, referred to as B1, B10 etc. MLwiN is not case-sensitive, so it will be most convenient for you to type in lower case although you may find it useful to adopt a convention of using capital letters and punctuation for annotating what you are doing.

Enhancements in Version 2.26

The following features are present in Version 2.26. For documentation, please see the separate 'MLwiN v2.26 manual supplement'

Estimation

- Predictions are now available for specified values of the explanatory variables as well as for the units in the data set
- There is a new method for estimating autocorrelated errors in continous time
- Ordinal variables can now be entered into the model as orthogonal polynomials
- There are extra features for data manipulation
- Features have been added to make the running of models from macros easier, including the ability to control the **Equations** window from a macro

Exploring, importing and exporting data

- Basic surface plotting with rotation is now available
- Model comparison tables showing estimates for the various models run can now be created and exported (for example to Word or Excel)
- SAS transport, SPSS, Stata and Minitab data files can now be saved and retrieved by MLwiN
- It is now possible to copy, paste and delete directly from the Names window

Improved ease of use

- The specification of models has been made easier, in particular, centring of explanatory variables, entering explanatory variables as polynomials and modifying explanatory variables already specified
- The open windows in MLwiN now appear as a row of tabs along the bottom
- Data can now be viewed by selecting variables from the Names window
- Specification of categorical variables has been made easier
- Column descriptors are now available to provide some information about variables
- MLwiN can now be invoked from the command line

MLwiN Help

The basic reference for MLwiN is provided by an extensive **Help** system. This uses the standard Windows Help conventions. Links are underlined and topics are listed under 'contents'. There is a principal **Help** button located on the main menu and context sensitive buttons located on individual screens. You can use the 'index' to search for a topic or alternatively if you click on the **find** tab you can search using keywords for the topic. Navigation through the Help system involves clicking on hypertext links, or using any of the options on the Help screen menu bars. You can also use any of the functions available under 'options' on the Windows Help toolbar, such as printing, etc.

Compatibility with existing MLn software

It is possible to use MLwiN in just the same way as MLn via the **Command interface** window. Opening this and clicking on the **Output** button allows you to enter commands and see the results in a separate window. For certain kinds of analysis this is the *only* way to proceed. MLwiN will read existing MLn worksheets, and a switch can be set when saving MLwiN worksheets so that they can be read by MLn. For details of all MLwiN commands see the relevant topics in the **Help** system. You can access these in the index by typing "command *name*" where *name* is the MLn command name.

Macros

MLwiN contains enhanced macro functions that allow users to design their own menu interfaces for specific analyses. A special set of macros for fitting discrete response data using quasilikelihood estimation has been embedded into the **Equations** window interface so that the fitting of these models is now entirely consistent with the fitting of Normal models. A full discussion of macros is given in the MLwiN **Help** system.

The structure of the User's Guide

Following this introduction the first chapter provides an introduction to multilevel modelling and the formulation of a simple model. A key innovative feature of MLwiN is the **Equations** window that allows the user to specify and manipulate a model using standard statistical notation. (This assumes that users of MLwiN will have a statistical background that encompasses a basic understanding of multiple regression analysis and the corresponding standard notation associated with that.) In the next chapter we introduce multilevel modelling by developing a multilevel model building upon a simple regression model. After that there is a detailed analysis of an educational data set that introduces the key features of MLwiN. Subsequent chapters take users through the analysis of different kinds of data, illustrating further features of MLwiN including its more advanced ones. The User's Guide concludes with two advanced chapters — on cross-classification models and multiple membership models — which describe how to fit these models using MLwiN commands.

We suggest that users take the time to work through at least the first tutorial to become familiar with the software. The **Help** system is extensive and provides full explanations of all MLwiN features and also offers help with many of the statistical procedures. Abridged versions of the tutorials are also available within the **Help** system.

Acknowledgements

The development of the MLwiN software has been the principal responsibility of Jon Rasbash and, more recently, Christopher Charlton, but also owes much to the efforts of a number of people outside the Centre for Multilevel Modelling.

Michael Healy developed the program NANOSTAT that was the precursor of MLn and hence MLwiN and we owe him a considerable debt for his inspi-

ration and continuing help. William Browne wrote the code for the MCMC modelling options with initial advice from David Draper. Geoff Woodhouse and Ian Plewis have contributed to earlier editions of the manual. Bob Prosser edited the manual, Amy Burch formatted previous versions in Word, and Mike Kelly converted the manual from Word to LaTeX.

The Economic and Social Research Council (ESRC) has provided continuous support to the Centre for Multilevel Modelling at the Institute of Education since 1986, and subsequently at the University of Bristol. Without their support MLwiN could not have happened. A number of visiting fellows have been funded by ESRC at various times: Ian Langford, Alastair Leyland, Toby Lewis, Dick Wiggins, Dougal Hutchison, Nigel Rice and Tony Fielding. They have contributed greatly.

Many others, too numerous to mention, have played their part and we particularly would like to acknowledge the stimulation and encouragement we have received from the team at the MRC Biostatistics unit in Cambridge and at Imperial College London. The BUGS software developments have complemented our own efforts. We are also most grateful to the Joint Information Systems Committee (U.K.) for funding a project related to parallel processing procedures for multilevel modelling.

Further information about multilevel modelling

There is a website that contains much of interest, including new developments, and details of courses and workshops. To view this go to the following address: `http://www.bristol.ac.uk/cmm/`. This website also contains the latest information about MLwiN software, including upgrade information, maintenance downloads, and documentation.

There is an active email discussion group about multilevel modelling. You can join this by sending an email to jiscmail@jiscmail.ac.uk with a single message line as follows: (Substituting your own first and last names for *firstname* and *lastname*)

Join multilevel firstname lastname

Technical Support

For MLwiN technical support please go to our technical support web page at `http://www.bristol.ac.uk/cmm/software/support/` for more details, including eligibility.

Chapter 1

Introducing Multilevel Models

1.1 Multilevel data structures

In the social, medical and biological sciences multilevel or hierarchically structured data are the norm and they are also found in many other areas of application. For example, school education provides a clear case of a system in which individuals are subject to the influences of grouping. Pupils or students learn in classes; classes are taught within schools; and schools may be administered within local authorities or school boards. The *units* in such a system lie at four different levels of a hierarchy. A typical multilevel model of this system would assign pupils to level 1, classes to level 2, schools to level 3 and authorities or boards to level 4. Units at one level are recognised as being grouped, or nested, within units at the next higher level.

In a household survey, the level 1 units are individual people, the level 2 units are households and the level 3 units, areas defined in different ways. Such a hierarchy is often described in terms of *clusters* of level 1 units within each level 2 unit etc. and the term *clustered population* is used.

In animal or child growth studies repeated measurements of, say, weight are taken on a sample of individuals. Although this may seem to constitute a different kind of structure from that of a survey of school students, it can be regarded as a 2-level hierarchy, with animals or children at level 2 and the set of measurement occasions for an individual constituting the level 1 units for that level 2 unit. A third level can be introduced into this structure if children are grouped into schools or young animals grouped into litters.

In trials of medical procedures, several centres may be chosen and individual patients studied within each one. Here the centres become the level 2 units and the patients the level 1 units.

In all these cases, we can see clear hierarchical structures in the population.

From the point of view of our models what matters is how this structure affects the measurements of interest. Thus, if we are measuring educational achievement, it is known that average achievement varies from one school to another. This means that students within a school will be more alike, on average, than students from different schools. Likewise, people within a household will tend to share similar attitudes etc. so that studies of, say, voting intention need to recognise this. In medicine it is known that centres differ in terms of patient care, case mix, etc. and again our analysis should recognise this.

1.2 Consequences of ignoring a multilevel structure

The point of multilevel modelling is that a statistical model explicitly should recognise a hierarchical structure where one is present: if this is not done then we need to be aware of the consequences of failing to do this.

In our first tutorial example we look at the relationship between an outcome or response variable which is the score achieved by 16 year old students in an examination and a predictor or explanatory variable which is a reading test score obtained by the same students just before they entered secondary school at the age of 11 years. The first variable is referred to as "exam score" and the second as "LRT score" where LRT is short for 'London Reading Test'. In the past it would have been necessary to decide whether to carry out this analysis at school level or at pupil level. Both of these single-level analyses are unsatisfactory, as we now show.

In a school-level or aggregate analysis, the mean exam score and the mean LRT score would first be calculated for each school. Ordinary regression would then be used to estimate a relationship between these. The main problem here is that it is far from clear how to interpret any relationship found. Any causal interpretation must include individual pupils, and information about these has been discarded. In practice it is possible to find a wide variety of models, each fitting the data equally well, but giving widely different estimates. Because of the difficulty of interpretation the results of such analyses depend on an essentially arbitrary choice of model. An empirical demonstration of this unreliability is given by Woodhouse & Goldstein (1989), who analyse examination results in English Local Education Authorities.

In a pupil-level analysis an average relationship between the scores would be estimated using data for all 4059 pupils. The variation between schools could be modelled by incorporating separate terms for each school. This procedure is inefficient, and inadequate for the purpose of generalisation. It is inefficient because it involves estimating many times more coefficients than

the multilevel procedure; and because it does not treat schools as a random sample it provides no useful quantification of the variation among schools in the population more generally.

By focusing attention on the levels of hierarchy in the population, multilevel modelling enables the researcher to understand where and how effects are occurring. It provides better estimates in answer to the simple questions for which single-level analyses were once used and in addition allows more complex questions to be addressed. For example Nuttall et al. (1989), using multilevel modelling, showed that secondary schools varied in the progress made by students from different ethnic groups: in some schools certain ethnic minority group children made more progress, in comparison with non-minority children, than in other schools.

Finally, carrying out an analysis that does not recognise the existence of clustering at all, for example a pupil level analysis with no school terms, creates serious technical problems. For example, ignored clustering will generally cause standard errors of regression coefficients to be underestimated. Consider also models of electoral behaviour. Voters are clustered within wards and wards within constituencies. If standard errors were underestimated it might be inferred, for example, that there was a real preference for one party or course of action over another when in fact that preference, estimated from the sample, could be ascribed to chance. Correct standard errors would be estimated only if variation at ward and constituency level were allowed for in the analysis. Multilevel modelling provides an efficient way of doing this. It also makes it possible to model and investigate the relative sizes and effects of ward characteristics and of constituency characteristics on electoral behaviour, as well as that of individual characteristics such as social group.

There is now a considerable literature on multilevel modelling, both theoretical and applied. The tutorial examples will enable the new user to become familiar with the basic concepts. More detailed discussions and statistical derivations can be found in the books by Bryk & Raudenbush (1992), Longford (1993) and Goldstein (2003).

1.3 Levels of a data structure

We shall use our exam data to illustrate the fundamental principle of multilevel modelling: the existence of different levels of variation. For this purpose schools will be the groups of interest.

We start by introducing some basic terminology and to keep matters simple we restrict attention to a single response variable and one predictor. We begin by looking at the data from a single school. In Figure 1.1 the exam scores of 73 children sampled from one of the schools in our sample are plotted against

the LRT scores for the same children. The relationship is summarised by the prediction or regression line.

Figure 1.1: Level 1 variation

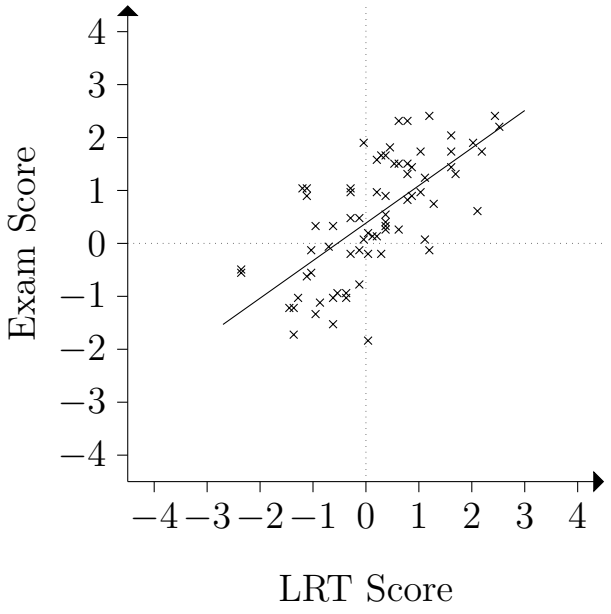
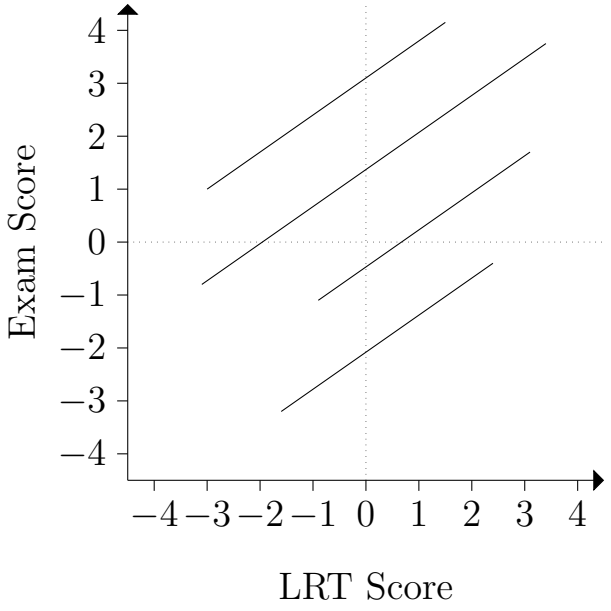


Figure 1.2: Level 2 variation in school summary lines



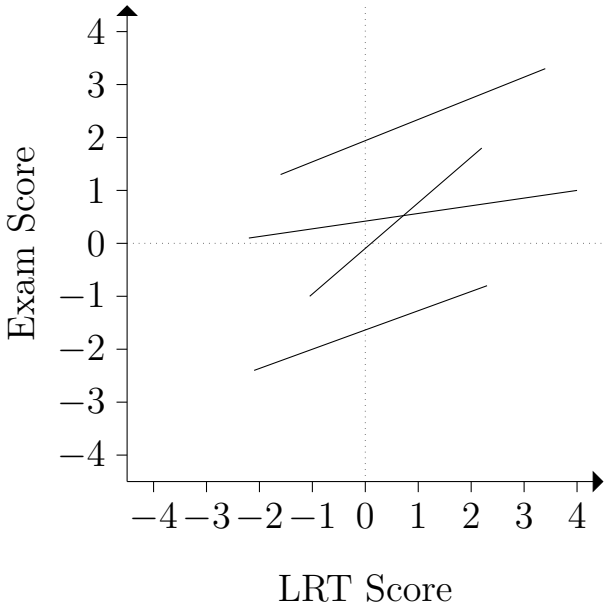
The lines in Figure 1.2 have different *intercepts*. The variation between these intercepts is called level 2 variation because in this example the schools are level 2 units. The schools are thought of as a random sample from a large underlying population of schools and 'school' is referred to as a *random classification*. The individual schools, like the individual pupils, are not of primary interest. Our interest is rather to make inferences about the variation among

all schools in the population, using the sampled schools as a means to that end.

If we regard the lines in Figure 1.2 as giving the prediction of the exam score for a given LRT score, then it is clear that the differences between schools is constant across the range of LRT scores. This kind of variation is known as *simple* level 2 variation.

If we allow the slopes of the lines to vary as in Figure 1.3, then the differences between the schools depend on the students' LRT scores. This is an example of *complex* level 2 variation.

Figure 1.3: Complex level 2 variation



Once again, the main focus of a multilevel analysis is not on the individual schools in the sample, but on estimating the pattern of variation in the underlying population of schools. Once this is done it becomes possible to attempt to explain the pattern in terms of general characteristics of schools, by incorporating further variables into the model. We can also obtain 'posterior' estimates of intercepts and slopes for each school and the procedures for this will be illustrated in the next chapter.

Before we set up a simple model for the examination data we briefly review the basic statistical theory of multilevel modelling. This section can be skipped by those familiar with the statistical background.

1.4 An introductory description of multilevel modelling

Figure 1.1 provided an illustration of level 1 variation for a single school, together with a regression line representing a summary relationship between the exam and LRT scores for the pupils in this school. The technique of 'ordinary least squares', or OLS, to produce this relationship is well known and provided by many computer packages, including MLwiN. Our interest, however, is to use the variation between all the schools of the sample in order to make inferences about the variation in the underlying population.

We use the regression line in order to revise some standard algebraic notation. The ordinary regression relationship for a single school may be expressed as:

$$y_i = a + bx_i + e_i \tag{1.1}$$

where subscript i takes values from 1 to the number of pupils in the school. In our example y_i and x_i are respectively the exam and LRT scores for the ith pupil. The regression relation can also be expressed as:

$$\hat{y}_i = a + bx_i \tag{1.2}$$

where $\hat{y}_i$ is the exam score predicted for the ith pupil by this particular summary relationship for the school. The intercept, a, is where the regression line meets the vertical axis and b is its slope. The expression $a+bx_i$ is known as the *fixed part* of the model.

In equation (1.1), e_i is the departure of the ith pupil's actual exam score from the predicted score. It is commonly referred to either as an *error* or as a *residual*. In this volume we shall use the latter term. This residual is that part of the score y_i which is not predicted by the fixed part regression relationship in equation (1.2). With only one school, the level 1 variation is just the variance of these e_i.

Turning now to the multilevel case of several schools which are regarded as a random sample of schools from a population of schools, we assume a regression relation for each school

$$\hat{y}_{ij} = a_j + bx_{ij} \tag{1.3}$$

Where the slopes are parallel and the subscript j takes values from 1 to the number of schools in the sample.

We can write the full model as

$$y_{ij} = a_j + bx_{ij} + e_{ij} \tag{1.4}$$

In general, wherever an item has two subscripts ij, it varies from pupil to pupil within a school. Where an item has a j subscript only it varies across schools but has the same value for all the pupils within a school. And where an item has neither subscript it is constant across all pupils and schools.

In a multilevel analysis the level 2 groups, in this case schools, are treated as a random sample from a population of schools. We therefore re-express equation (1.3) as

$$\begin{aligned} a_j &= a + u_j \\ \hat{y}_{ij} &= a + bx_{ij} + u_j \end{aligned} \tag{1.5}$$

Where a, with no subscripts, is a constant and u_j, the departure of the j^{th} school's intercept from the overall value, is a level 2 residual which is the same for all pupils in school j.

The model for actual scores can now be expressed as:

$$y_{ij} = a + bx_{ij} + u_j + e_{ij} \tag{1.6}$$

In this equation, both u_j and e_{ij} are random quantities, whose means are equal to zero; they form the *random part* of the model (1.6). We assume that, being at different levels, these variables are uncorrelated and we further make the standard assumption that they follow a Normal distribution so that it is sufficient to estimate their variances, σ_u^2 and σ_e^2 respectively. The quantities a and b, the mean intercept and slope, are fixed and will also need to be estimated.

It is the existence of the two random variables u_j and e_{ij} in equation (1.6) that marks it out as a multilevel model. The variances σ_u^2 and σ_e^2 are referred to as *random parameters* of the model. The quantities a and b are known as *fixed parameters*.

A multilevel model of this simple type, where the only random parameters are the intercept variances at each level, is known as a variance components model. In order to specify more general models we need to adapt the notation of (1.6). First, we introduce a special explanatory variable x_0, which takes the value 1 for all students.

This allows every term on the right hand side of (1.6) to be associated with an explanatory variable. Secondly, we use β_0, β_1 etc. for the fixed parameters, the subscripts 0, 1 etc. matching the subscripts of the explanatory variables to which they are attached. Similarly, we incorporate a subscript 0 into the random variables and write

$$y_{ij} = \beta_0 x_0 + \beta_1 x_{1ij} + u_{0j} x_0 + e_{0ij} x_0 \tag{1.7}$$

Finally, we collect the coefficients together and write

$$\begin{aligned} y_{ij} &= \beta_{0ij} x_0 + \beta_1 x_{1ij} \\ \beta_{0ij} &= \beta_0 + u_{0j} + e_{0ij} \end{aligned} \tag{1.8}$$

Thus we have specified the random variation in y in terms of random coefficients of explanatory variables. In the present case the coefficient of x_0 is random at both level 1 and level 2. The zero subscripts on the level 1 and level 2 random variables indicate that these are attached to x_0.

For the standard model we assume that the response variable is normally distributed and this is usually written in standard notation as follows

$$y \sim \mathrm{N}(XB, \Omega) \tag{1.9}$$

where XB is the fixed part of the model and in the present case is a column vector beginning:

$$\begin{pmatrix} \beta_0 x_{011} + \beta_1 x_{111} \\ \beta_0 x_{021} + \beta_1 x_{121} \\ \cdots \end{pmatrix}$$

The symbol Ω represents the variances and covariances of the random terms over all the levels of the data. In the present case it denotes just the variances at level 1 and level 2. Equations (1.8) and (1.9) form a complete specification for our model and MLwiN uses this notation to specify the models that are described in the following chapters.

Chapter 2

Introduction to Multilevel Modelling

One aim of this chapter is to demonstrate how multilevel modelling builds on traditional statistical methods for the comparison of groups where, for example, the groups may be boys and girls or different schools. We begin with an overview of standard regression methods for comparing the means of two or more groups, commonly called analysis of variance (ANOVA) or sometimes 'fixed effects' models. We then contrast this approach with multilevel or 'random effects' modelling. The chapter also provides a revision of methods for single-level statistical inference, including Normal tests for comparing means and likelihood ratio tests, which are also used in multilevel modelling.

The other aim of the chapter is to provide an introduction to the MLwiN software. The chapter is a tutorial which will take you through procedures for manipulating data, carrying out descriptive analysis, creating graphs, specifying and estimating ordinary least squares (OLS) regression and multilevel models, and making inferences.

2.1 The tutorial data set

For illustration here, we use an educational data set for which an MLwiN worksheet has already been prepared. Usually, at the beginning of an analysis, you will have to create such a worksheet yourself either by entering the data directly or by reading a file or files prepared elsewhere. Facilities for doing this are described in Chapter 8. The data in the worksheet we use have been selected from a very much larger data set of examination results from six inner London Education Authorities (school boards). A key aim of the original analysis was to establish whether some schools were more 'effective' than others in promoting students' learning and development, taking account of variations in the characteristics of students when they started

secondary school. The analysis then looked for factors associated with any school differences found. Thus the focus was on an analysis of examination performance after adjusting for student intake achievements. As you explore MLwiN in this and following chapters using the simplified data set you will also be imitating, in a simplified way, the procedures of the original analysis. For a full account of that analysis see Goldstein et al. (1993).

The data set contains the following variables:

Variable	Description
School	Numeric school identifier
Student	Numeric student identifier
Normexam	Student's exam score at age 16, normalised to have approximately a standard Normal distribution. (Note that the normalisation was carried out on a larger sample, so the mean in this sample is not exactly equal to 0 and the variance is not exactly equal to 1.)
Cons	A column of ones. If included as an explanatory variable in a regression model, its coefficient is the intercept. See Chapter 7.
Standlrt	Student's score at age 11 on the London Reading Test, standardised using Z-scores.
girl	1 = girl, 0 = boy
schgend	School's gender (1 = mixed school, 2 = boys' school, 3 = girls' school)
Avslrt	Average LRT score in school
Schav	Average LRT score in school, coded into 3 categories (1 = bottom 25%, 2 = middle 50%, 3 = top 25%)
Vrband	Student's score in test of verbal reasoning at age 11, coded into 3 categories (1 = top 25%, 2 = middle 50%, 3 = bottom 25%)

Note that in order to fit a model with n hierarchical levels MLwiN requires your data to be sorted by level 1 nested within level 2 within level 3 ... level n. For example here the data are sorted by students (level 1) within schools (level 2). There is a sort function available from the ***Data Manipulation*** *menu.*

2.2 Opening the worksheet and looking at the data

When you start MLwiN the main window appears. Immediately below the MLwiN title bar is the **menu bar** and below it the **tool bar** as shown:

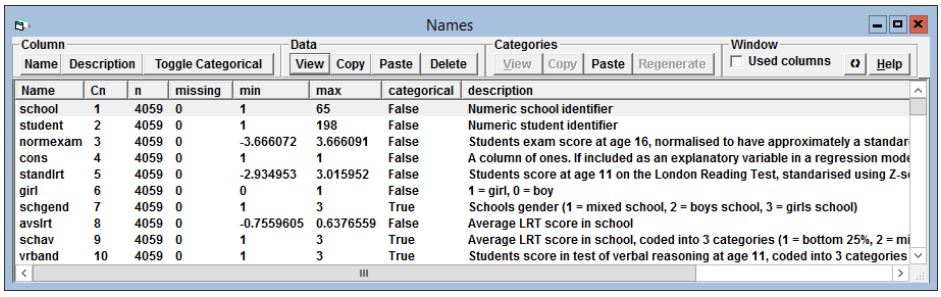
File Edit Options Model Estimation Data Manipulation Basic Statistics Graphs Window Help
Start More Stop IGLS
Estimation control..

These menus are fully described in the online **Help** system. This may be accessed either by clicking the **Help** button on the menu bar shown above or (for context-sensitive **Help**) by clicking the **Help** button displayed in the window you are currently working with. You should use this system freely.

The buttons on the tool bar relate to model estimation and control, and we shall describe these in detail later. Below the tool bar is a blank workspace into which you will open windows. These windows form the rest of the graphical user interface that you use to specify tasks in MLwiN. Below the workspace is the **status bar**, which monitors the progress of the iterative estimation procedure. Open the tutorial worksheet as follows:

- Select **File** menu
- Select **Open worksheet**
- Select **tutorial.ws**
- Click **Open**

When this operation is complete the filename will appear in the title bar of the main window and the status bar will be initialised. The **Names** window will also appear, giving a summary of the data in the worksheet:



Name	Cn	n	missing	min	max	categorical	description
school	1	4059	0	1	65	False	Numeric school identifier
student	2	4059	0	1	198	False	Numeric student identifier
normexam	3	4059	0	-3.666072	3.666091	False	Students exam score at age 16, normalised to have approximately a standar
cons	4	4059	0	1	1	False	A column of ones. If included as an explanatory variable in a regression mod
standlrt	5	4059	0	-2.934953	3.015952	False	Students score at age 11 on the London Reading Test, standarised using Z-s
girl	6	4059	0	0	1	False	1 = girl, 0 = boy
schgend	7	4059	0	1	3	True	Schools gender (1 = mixed school, 2 = boys school, 3 = girls school)
avslrt	8	4059	0	-0.7559605	0.6376559	False	Average LRT score in school
schav	9	4059	0	1	3	True	Average LRT score in school, coded into 3 categories (1 = bottom 25%, 2 = mi
vrband	10	4059	0	1	3	True	Students score in test of verbal reasoning at age 11, coded into 3 categories

The MLwiN worksheet holds the data and other information in a series of columns, as on a spreadsheet. These are initially named c1, c2, ..., but we recommend that they be given meaningful names to show what their contents relate to. This has already been done in the **tutorial** worksheet that you have loaded.

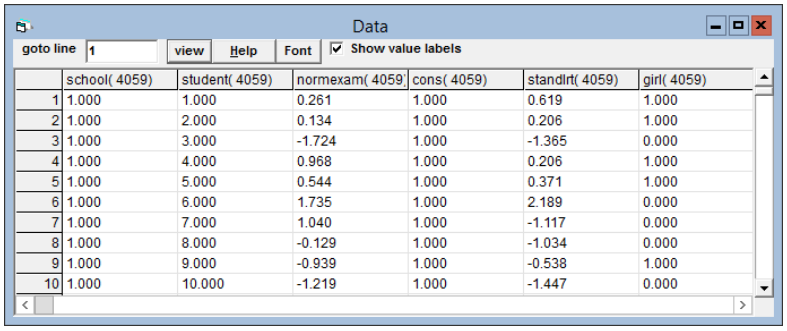
Each line in the body of the **Names** window summarises a column of data. In the present case only the first 10 of the 1500 columns of the worksheet contain data. Each column contains 4059 values, one for each student represented in the data set. There are no missing values, and the minimum and maximum value in each column are shown.

*Note the **Help** button on the window's tool bar. We shall see what some of the other buttons do later in this manual; the rest are documented in the MLwiN v2.10 Manual Supplement.*

You can view individual values in the data using the **Data** window as follows:

- On the **Data Manipulation** menu, select **View or edit data**

The following window appears:



	school(4059)	student(4059)	normexam(4059)	cons(4059)	standlrt(4059)	girl(4059)
1	1.000	1.000	0.261	1.000	0.619	1.000
2	1.000	2.000	0.134	1.000	0.206	1.000
3	1.000	3.000	-1.724	1.000	-1.365	0.000
4	1.000	4.000	0.968	1.000	0.206	1.000
5	1.000	5.000	0.544	1.000	0.371	1.000
6	1.000	6.000	1.735	1.000	2.189	0.000
7	1.000	7.000	1.040	1.000	-1.117	0.000
8	1.000	8.000	-0.129	1.000	-1.034	0.000
9	1.000	9.000	-0.939	1.000	-0.538	1.000
10	1.000	10.000	-1.219	1.000	-1.447	0.000

When this window is initially opened, it always shows the first columns containing data in the worksheet. The exact number of values shown depends on the space available on your screen. You can view any selection of columns, spreadsheet fashion, as follows:

- Click the **View** button
- Select columns to view
- Click **OK**

Alternatively, in version 2.10 you can select the columns you wish to view in the **Names** window and then click the **Data** button at the top — this will bring up the **Data** window displaying the selected columns.

You can select a block of adjacent columns either by pointing and dragging or by selecting the column at one end of the block and holding down 'Shift' while you select the column at the other end. You can add to an existing selection by holding down 'Ctrl' while you select new columns or blocks. Use the scroll bars of the **Data** window to move horizontally and vertically through the data, and move or resize the window if you wish. You can go straight to line 1035, for example, by typing 1035 in the **goto line** box, and you can highlight a particular cell by pointing and clicking. This provides a means to edit data.

Having viewed your data you will typically wish to tabulate and plot selected variables, and derive other summary statistics, before proceeding to actual

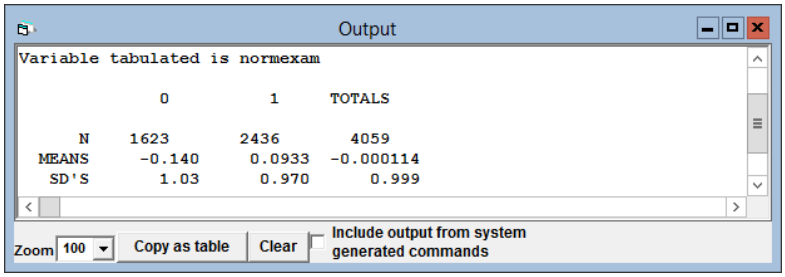
modelling. Tabulation and other basic statistical operations are available on the **Basic Statistics** menu. These operations are described in the **Help** system.

2.3 Comparing two groups

Suppose that we wish to examine the relationship between exam performance at age 16 and gender. We could begin by comparing the mean of **normexam** for boys and girls:

- From the **Basic Statistics** menu, select **Tabulate**
- Under **Output Mode**, select **Means**
- From the drop-down list next to **Variate column**, select **normexam**
- From the drop-down list next to **Columns**, select **girl**
- Click **Tabulate**

You should obtain the following output:



On average, the girls (coded 1) in the sample performed better than the boys (coded 0), with a mean difference of $0.093 - (-0.140) = 0.233$. From the standard deviations, it can also be seen that boys' scores are slightly more variable than girls'. To begin with, we assume that the variability in exam scores is the same for boys and girls but we will modify this assumption in Chapter 7. The Totals column of the table gives the overall mean and the pooled standard deviation (s_P), calculated by pooling the standard deviations for girls and boys.

While this descriptive analysis provides useful information about our sample of 4059 students, of major interest is the population of students from which this sample was drawn. To test whether there is a gender difference in the mean exam score *in the population*, we would traditionally carry out a two-sample t-test. In the present case, as in all the examples considered in this manual, we use 'large sample' procedures for significance tests and confidence intervals. Thus, instead of a t-test we use a Normal distribution test, and

more generally likelihood ratio tests. The test statistic for comparing the mean exam score for boys and girls is:

$$Z = \frac{\bar{X}_G - \bar{X}_B}{\sqrt{s_P^2\left(\frac{1}{n_G} + \frac{1}{n_B}\right)}} = \frac{0.093 - (-0.14)}{\sqrt{0.992^2\left(\frac{1}{2436} + \frac{1}{1623}\right)}} = \frac{0.233}{0.032} = 7.23$$

This value may be compared with a standard Normal distribution (with mean zero and standard deviation 1) to obtain a p-value, which is the probability of obtaining a test statistic as or more extreme than 7.23 if the null hypothesis were true. Although a value as high as 7.23 is clearly significant, we demonstrate how a p-value may be computed in MLwiN:

- From the **Basic Statistics** menu, select **Tail Areas**
- Under **Operation**, select **Standard Normal distribution**
- In the empty box next to **Value**, type the value of the test statistic, i.e, **7.23**
- Click **Calculate**

You should obtain a value of $2.415e-013 = 2.415 \times 10^{-13}$. We double this value to obtain the p-value for a two-sided test. (A two-sided test is appropriate since we did not specify *a priori* the direction of any gender difference.) The p-value is extremely small, which implies that we are highly unlikely to obtain a test statistic as extreme as 7.23 if there was in fact no difference between boys and girls in the population. We therefore conclude that there is a real population gender difference in the mean of **normexam**. We can state that the effect of gender is "statistically significant" at a very high level of significance.

We could also calculate a confidence interval for the population mean difference between girls and boys, $\mu_G - \mu_B$. The 95% confidence interval is

$$(\bar{X}_G - \bar{X}_B) \pm 1.96\ SE(\bar{X}_G - \bar{X}_B) = 0.233 \pm 1.96(0.032) = (0.170, 0.296)$$

The true mean difference is unlikely to lie outside these limits.

An alternative, but equivalent, approach to the Normal test is to fit a regression model in which we allow **normexam** to depend on gender. The regression model is fitted using ordinary least squares and is often referred to as an OLS model. When the explanatory variable is categorical, the model is more commonly called an Analysis of Variance (ANOVA) model. As we shall see, the advantage of using this approach is that it can be extended to

compare more than two groups and to allow for the effects of other explanatory variables. A regression model for comparing the mean of **normexam** for boys and girls can be written:

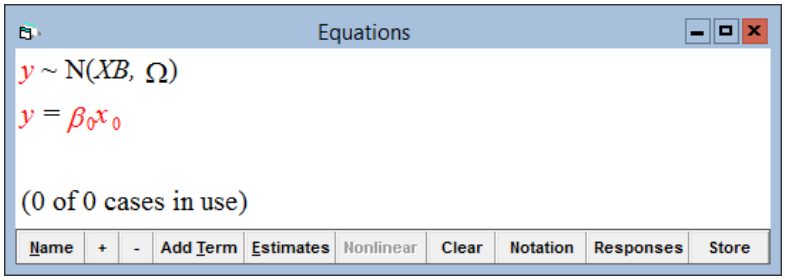
$$y_i = \beta_0 + \beta_1 x_i + e_i \tag{2.1}$$

where y_i is the value of **normexam** for student i, and x_i is their gender (0 for a boy, and 1 for a girl). The parameter β_0 is called the "intercept", which in this case represents the overall mean of **normexam** for boys (i.e, the mean of y when x=0). The parameter β_1 represents the effect of gender, specifically the difference between boys and girls in the mean of **normexam**. The term e_i, known as the residual (or error) term, is the difference between the observed value of **normexam** for student i and their value predicted by the regression, i.e, the population mean for students of the same gender.

To specify a regression model in MLwiN:

- From the **Model** menu, select **Equations**

The following window will appear:

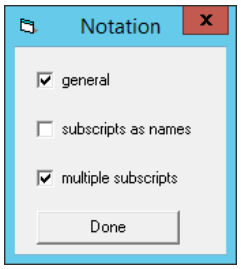


The **Equations** window is used to specify statistical models and to view the results of fitting those models. The **Equations** window has a number of different display modes. Two of these modes are simple or general notation. General notation is the default mode. Since this is an introductory chapter, we will switch to simple notation in order to make the transition between single-level models and multilevel models easier to follow.

You can change the appearance of the **Equations** window in several other ways using the buttons at the bottom of the window. To have variable names displayed in place of x, y etc., you can click the **Name** button. To see variable names for subscripts instead of i, j etc., you can use the **Notation** button, and to show numerical results rather than mathematical symbols you can click the **Estimates** button. To request that the full model specification is displayed, click the + button, and to suppress this specification, click the **-** button. We shall demonstrate each of these display modes as we go through the chapter:

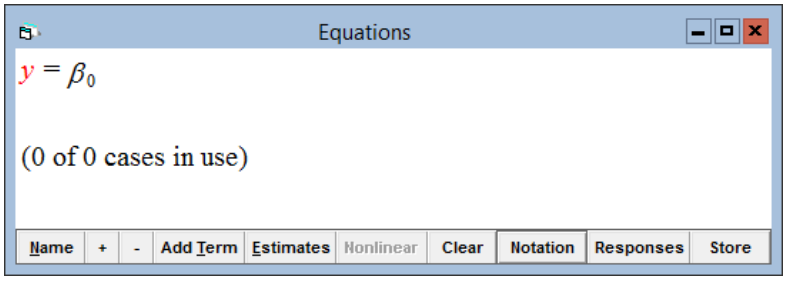
- Click the **Notation** button on the bottom of **Equations** window

The following window will appear:



- Clear the **general** tick box
- Click the **Done** button

The Equations window now looks as follows:

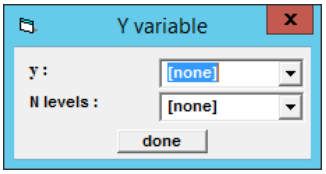


Note that y is shown in red indicating that it has not yet been defined.

To define the response variable:

- Click on the red **y** in the **Equations** window

The following window will appear:



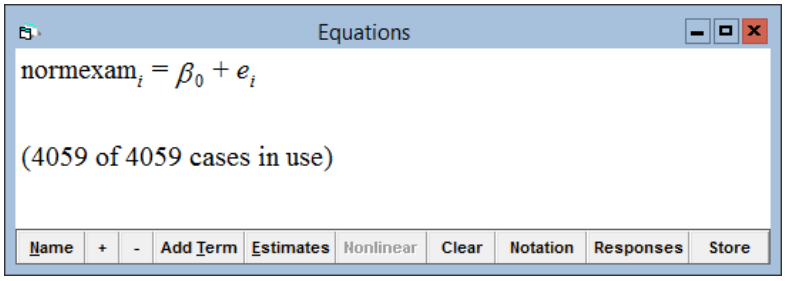
- From the drop-down list labelled **y:**, select **Normexam**
- From the drop-down list labelled **N levels:**, select **1-i**

After you have made a selection for **N levels:**, the Y variable window expands to allow you to specify the variable(s) containing identification codes for units at each level:

- From the drop-down list labelled **level 1(i):**, select **student**
- Click **done**

Note that level 1 corresponds to the level at which the response measurements were made.

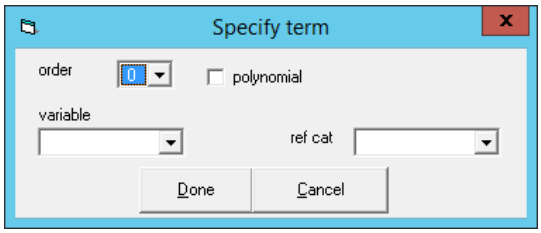
The **Equations** window now looks like this:



Now we need to add the gender variable:

- Click the **Add Term** button

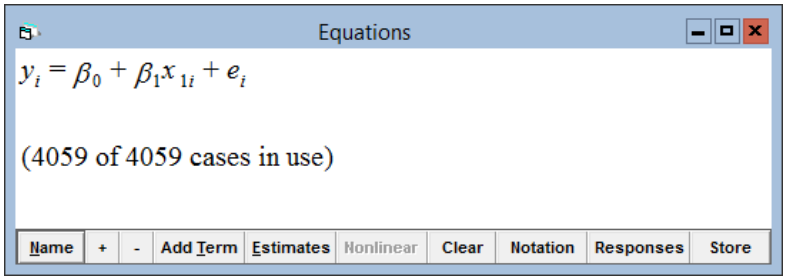
This brings up the following window:



This window allows you to add continuous and categorical variables and higher level interactions between variables. To add the gender variable:

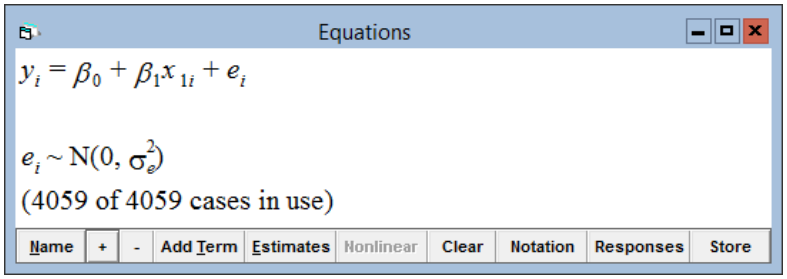
- Select **girl** from the drop-down list labelled **variable**
- Click **Done**

The **Equations** window will now look like this:


Equations
$y_i = \beta_0 + \beta_1 x_{1i} + e_i$
(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

Girl is a categorical variable coded 0 for boys and 1 for girls. A (0,1) variable such as this is often called a *dummy variable*. If **girl** = 0, then from the regression equation above, the mean of **normexam** is β_0, so β_0 is the mean for boys. For girls (**girl** = 1), the mean of **normexam** is $\beta_0 + \beta_1$, so β_1 is the girls' mean minus the boys' mean. The category coded 0 is called the *reference category*.

We can change the display mode so that we view mathematical symbols (e.g. y, x) rather than variable names. If we click on the **Name** button in the **Equations** window, we see the exact form of equation (2.1).

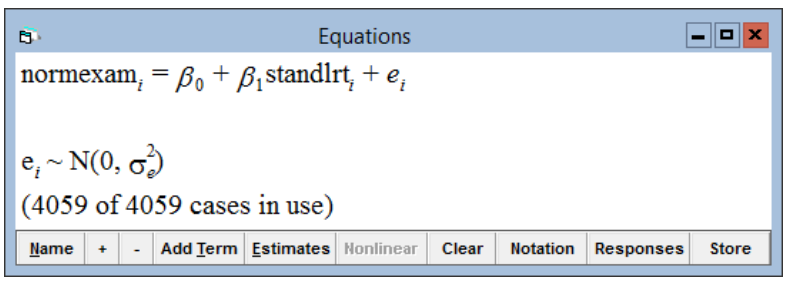

Equations
$y_i = \beta_0 + \beta_1 x_{1i} + e_i$
$e_i \sim N(0, \sigma_e^2)$
(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

Note that if you clicked on **Name** *again the display would revert to variable names. The* **Name** *button is called a 'toggle' since it allows us to switch between display modes.*

We can also choose the amount of detail in the model specification that we would like to be displayed in the **Equations** window. Currently, we have only the most basic information about the model — the linear regression equation. Using the + button we can request further details:

- Click the + button on the **Equations** window twice

The **Equations** window now displays:


Equations
$\text{normexam}_i = \beta_0 + \beta_1 \text{standlrt}_i + e_i$
$e_i \sim N(0, \sigma_e^2)$
(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The second line tells us that the residuals are assumed to be Normally distributed with mean 0 and variance σ_e^2.

> *Note that we could suppress this line in the model specification by clicking on the − button. Like the* **Name** *button the + and − buttons are toggles which allow us to switch between display modes.*

We shall now get MLwiN to estimate the parameters of the model specified in the previous section. We are going to estimate the two parameters β_0 and β_1 which constitute the *fixed* part of the model, and the variance of the residuals, σ_e^2. The residuals and their variance are referred to as the *random* part of the model.

To see the status of the model (i.e, whether it has been fitted or not), we can use the **Estimates** button. The **Estimates** button allows us to choose between three display modes for the parameters:

1. Mathematical symbols in black typeface (the default)
2. Mathematical symbols with colour indicating whether the model has been fitted
3. Numerical results after a model fit.

Like the **Name**, + and − buttons, the **Estimates** button allows us to toggle between different display modes. So if we start at the default mode and click **Estimates** twice we will see the numerical results, and if we click **Estimates** once more we return to the default:

- Click the **Estimates** button on the **Equations** window toolbar

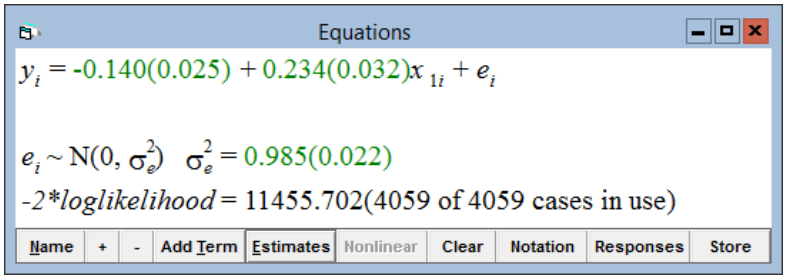
You should see highlighted in blue the parameters that are to be estimated. To begin the estimation we use the tool bar of the main MLwiN window. The **Start** button starts estimation, the **Stop** button stops it, and the **More** button resumes estimation after a stop:

- Click **Start**

The parameters will now turn green, indicating that the model has been fitted.

- Click **Estimates** again

You will now see the parameter estimates displayed with their standard errors in brackets:


Equations
$y_i = -0.140(0.025) + 0.234(0.032)x_{1i} + e_i$
$e_i \sim N(0, \sigma_e^2)$ $\sigma_e^2 = 0.985(0.022)$
-2*loglikelihood = 11455.702(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

Once a model has been fitted, another line appears at the bottom of the display giving the value for a log-likelihood function. This value can be used in the comparison of two different models. This will be discussed later in this chapter.

From the estimated model, we see that the population mean of **normexam** for girls is estimated to be 0.234 units higher than the mean for boys. This is the difference between the sample means of **normexam** for girls and boys which we obtained earlier, i.e, $0.093 - (-0.140) = 0.233$. The intercept estimate of -0.140 corresponds to the sample mean for boys; the predicted means, which we obtain from the model for each gender, match the sample means.

> *Note also that the estimated residual* $\hat{\sigma}_e^2 = 0.985$ *is equal to the sample variance of the exam scores after pooling across gender:* $s_p^2 = 0.992^2 = 0.985$ *(a circumflex over a term means "estimate of").*

The statistical model also provides us with standard errors, which allow us to make population inferences. In particular, we can test whether there is a gender difference in the population mean of **normexam**. The null hypothesis of no gender difference may be expressed in terms of the model parameters as $H_0 : \beta_1 = 0$. The test statistic for the Normal test is calculated as $\hat{\beta}_1/\mathrm{SE}(\hat{\beta}_1) = 0.234/0.032 = 7.31$ which, apart from rounding errors, is very close to the value obtained from the standard two-sample Normal test.

2.4 Comparing more than two groups: Fixed effects models

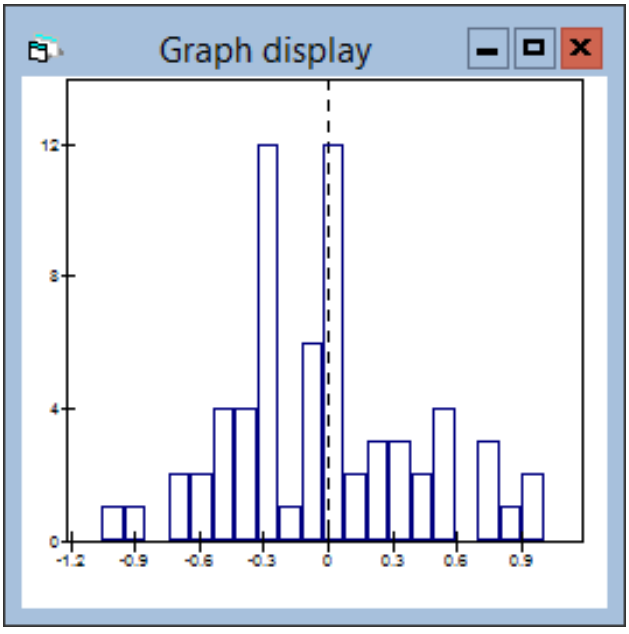
In the last section, we saw how the means of two groups can be compared using a regression model. Often, however, we wish to compare more than two groups. For example, we might wish to compare exam performance across schools. It is straightforward to modify the regression model in equation (2.1) to allow comparisons among multiple groups.

Before considering the model for comparing more than two groups, we conduct a descriptive analysis. To obtain the mean of **normexam** for each of the 65 schools in the sample:

- From the **Basic Statistics** menu, select **Tabulate**
- Under **Output Mode**, select **Means**
- For **Variate column**, select **normexam** from the drop-down menu
- For **Columns**, select **school**
- Check the box next to **Store in**, then select **C15** from the drop-down menu directly below
- Click **Tabulate**

You will obtain an output window containing the sample size, and mean and standard deviation of **normexam**, for each school. The means are stored in column C15 of the worksheet. When there are a large number of groups to be compared, as here, it is helpful to display the distribution graphically using a histogram. To obtain a histogram of the school means of **normexam**:

- From the **Graphs** menu, select **Customised Graph(s)**
- Next to **y**, select **C15** from the pull-down menu
- Under **plot type**, select **histogram**
- Click **Apply**



The histogram should look like the above figure.

From the histogram, we see that there is a large amount of variation in the mean of **normexam** across schools.

One way to describe the variation in the mean of **normexam** across schools is to fit a regression model, which includes a series of dummy variables for schools. For each of the 65 schools, we can define a dummy or indicator variable as follows:

$$\begin{aligned} x_j &= 1 \text{ for school } j \\ &= 0 \text{ otherwise} \\ &\text{for } j = 1, 2, \ldots, 65 \end{aligned}$$

In fact, we only need to know a student's value on 64 of these dummy variables to determine which school they attended. For example, if we know that x_j= 0 for j= 1, 2, ..., 64, then we can infer that the student attended school 65. We therefore only need to include 64 of the dummy variables in the model, and the school corresponding to the variable which is left out is the reference school to which all other schools are compared. If we take school 65 as the reference, a regression model, which allows for differences between schools can be written:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_{64} x_{64i} + e_i \tag{2.2}$$

where the coefficient β_j of dummy variable x_j represents the difference between the mean of **normexam** for school j and the mean for school 65, and β_0 is the mean for school 65. (Alternatively, we could allow for school effects by including the full set of 65 dummy variables and excluding the intercept term β_0. This would allow us to recover the school means directly, but such a model can only be fitted using *general notation* - see Chapter 7.)

Equation (2.2) represents what is commonly known as an analysis of variance (ANOVA) model, also known as a *fixed effects* model for reasons which shall be discussed in the next section. From (2.2), the mean for school 1 is $\beta_0 + \beta_1$, and the mean for school 2 is $\beta_0 + \beta_2$. In general, the mean for school j ($j \neq 65$) is $\beta_0 + \beta_j$. The mean for school 65 (the reference school) is β_0. Therefore, the ANOVA model is more commonly written in the following equivalent form:

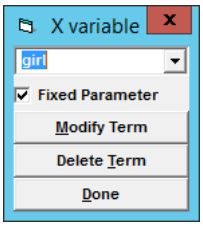
$$\begin{aligned} y_{ij} &= \beta_0 + \beta_j + e_{ij} \\ e_{ij} &\sim \mathrm{N}(0, \sigma_e^2) \end{aligned} \tag{2.3}$$

In this specification, y_{ij} is the value of **normexam** for student i in school j. j ranges over $1 \ldots 65$. β_{65} has a value of 0.

The model in equation (2.2) can be specified in MLwiN as follows:

- In the **Equations** window, click on the **girl** (or $\mathbf{x_{1i}}$) term

The following window appears:



- Click **delete Term**

The next step is to define **school** as a categorical variable. To do this:

- From the **Data Manipulation** menu, select **Names**
- Highlight **school**, then click on the **Toggle Categorical** button
- Click the **View** button in the categories section

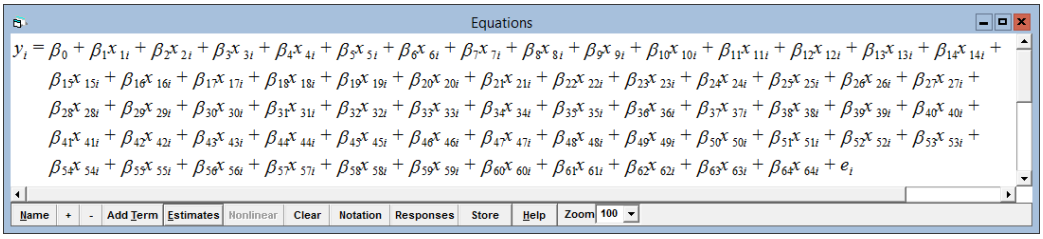
A label has been given to each category. By default this will be the column name concatenated with the category code, i.e. school_1, school_2 etc.

- To accept the defaults, click **OK**
- Close the **Names** window

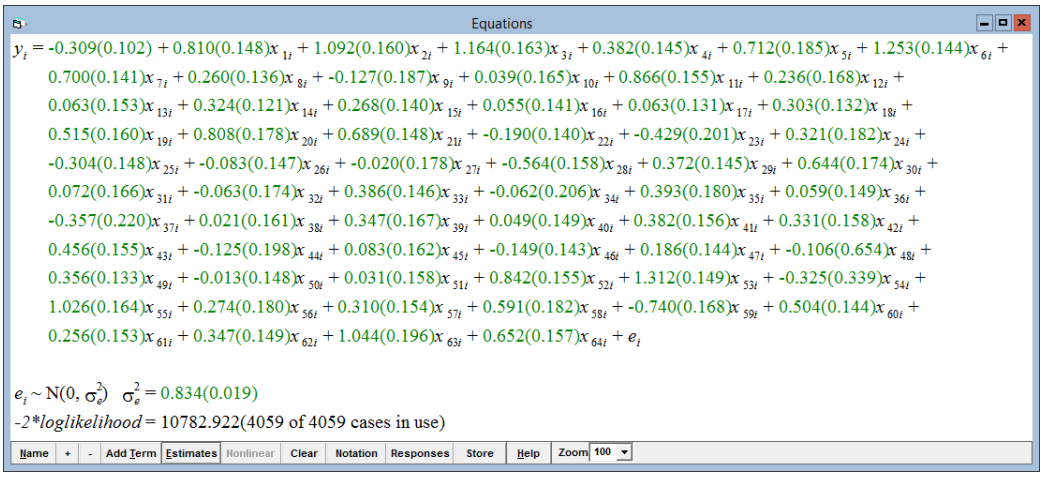
Having defined **school** as categorical, we can now add the school dummy variables to the model. By default, the lowest category number of a categorical variable is taken as the reference category. In the current case this would contrast schools 2 to 65 with school 1 (the default). We can, however, change the reference category. To make school 65 the reference school when we add in the dummy variables, do the following:

- Click on the **Add Term** button in the **Equations** window
- In the window that appears, select **school** from the **variable** drop-down list
- Select **school_65** from the **reference category** drop-down list
- Click **Done**

The **Equations** window will now show model (2.2):


Equations
$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{4i} + \beta_5 x_{5i} + \beta_6 x_{6i} + \beta_7 x_{7i} + \beta_8 x_{8i} + \beta_9 x_{9i} + \beta_{10} x_{10i} + \beta_{11} x_{11i} + \beta_{12} x_{12i} + \beta_{13} x_{13i} + \beta_{14} x_{14i} +$
$\beta_{15} x_{15i} + \beta_{16} x_{16i} + \beta_{17} x_{17i} + \beta_{18} x_{18i} + \beta_{19} x_{19i} + \beta_{20} x_{20i} + \beta_{21} x_{21i} + \beta_{22} x_{22i} + \beta_{23} x_{23i} + \beta_{24} x_{24i} + \beta_{25} x_{25i} + \beta_{26} x_{26i} + \beta_{27} x_{27i} +$
$\beta_{28} x_{28i} + \beta_{29} x_{29i} + \beta_{30} x_{30i} + \beta_{31} x_{31i} + \beta_{32} x_{32i} + \beta_{33} x_{33i} + \beta_{34} x_{34i} + \beta_{35} x_{35i} + \beta_{36} x_{36i} + \beta_{37} x_{37i} + \beta_{38} x_{38i} + \beta_{39} x_{39i} + \beta_{40} x_{40i} +$
$\beta_{41} x_{41i} + \beta_{42} x_{42i} + \beta_{43} x_{43i} + \beta_{44} x_{44i} + \beta_{45} x_{45i} + \beta_{46} x_{46i} + \beta_{47} x_{47i} + \beta_{48} x_{48i} + \beta_{49} x_{49i} + \beta_{50} x_{50i} + \beta_{51} x_{51i} + \beta_{52} x_{52i} + \beta_{53} x_{53i} +$
$\beta_{54} x_{54i} + \beta_{55} x_{55i} + \beta_{56} x_{56i} + \beta_{57} x_{57i} + \beta_{58} x_{58i} + \beta_{59} x_{59i} + \beta_{60} x_{60i} + \beta_{61} x_{61i} + \beta_{62} x_{62i} + \beta_{63} x_{63i} + \beta_{64} x_{64i} + e_i$
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

After clicking on the **Start** button to run the model, you should obtain the following results (press the **Estimates** button to see the numerical results):



From the model, the predicted mean of **normexam** for the reference school 65 is $\hat{\beta}_0 = -0.309$. We can compare this to the sample mean of **normexam** in school 65, stored in c15:

- From the **Data Manipulation** menu, select **View or edit data**
- Click on **View**, select **C15**, and scroll down to row 65

The sample mean for school 65 is indeed -0.309.

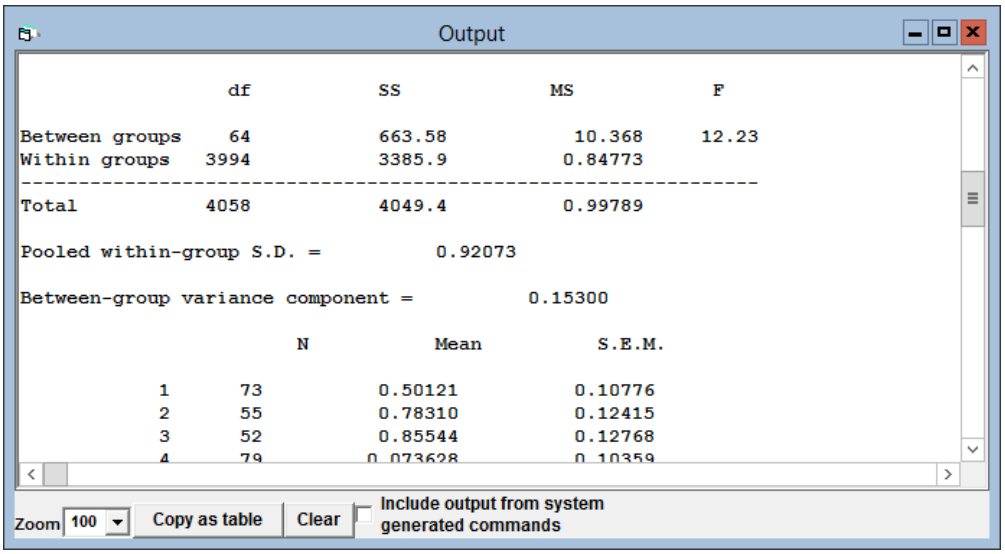
The model also provides estimated differences in the population mean of **normexam** for any pair of schools. For example, the estimated difference between schools 1 and 65 (the reference school) is $\hat{\beta}_1 = 0.810$, and the estimated difference between schools 1 and 2 is $\hat{\beta}_1 - \hat{\beta}_2 = 0.810 - 1.092 = -0.282$. These values correspond to the differences between the sample means given in column C15.

We can test for a difference between any school j and school 65 by carrying out a Normal test on the difference parameter β_j. We can also carry out a global test for school differences, i.e. a test of the null hypothesis $H_0 : \beta_1 = \beta_2 = \cdots = \beta_{64} = 0$ (no differences between schools). Traditionally an F-test would be used to test for differences between the school means. The test statistic for an F-test is based on the sum of squares of differences between schools and the sum of squares of differences between students within schools,

which are usually displayed in the form of an analysis of variance table. To obtain an ANOVA table in MLwiN:

- From the **Basic Statistics** menu, select **One way ANOVA**
- In the **Response** list, select **normexam**
- In the **Group codes** list, select **school**

This will produce the following output in the **Output** window:



The p-value for the test of differences between schools is found by comparing the F-statistic, 12.23, with a F-distribution on 64 and 3994 degrees of freedom. To do this in MLwiN:

- From the **Basic Statistics** menu, select **Tail Areas**
- Under **Operation**, select **F distribution**
- In the empty box next to **Value**, type **12.23**
- Next to **Degrees of freedom**, type **64**
- Next to **Denominator**, type **3994**
- Click **Calculate**

The p-value is very small, so we conclude that there are significant differences between schools. The 'within schools mean square' in the ANOVA table is 0.848; this is the estimate of the residual variance σ_e^2. This estimate is different from the value of 0.834 given in the **Equations** window because of a difference in the estimation algorithm used[1]. The regression model gives

[1] ANOVA uses restricted iterative generalised least squares (RIGLS), while models specified via the **Equations** window use iterative generalised least squares (IGLS) by default. The difference between IGLS and RIGLS is described in the **Help** system. The estimation method can be changed to RIGLS from the **Options** menu.

us estimates of the difference between the means for schools 1 to 64 and school 65. For example, the difference between school 1 and school 65 is 0.810. Looking at the ANOVA output, which lists school means directly, we see that we get identical estimates.

For 'large' samples, an alternative to the F-test for group comparisons is the *likelihood ratio test*. The likelihood ratio test is used to compare two "nested" models. Two models are considered nested if one model can be thought of as a restricted form of the other[2]. To test $H_0 : \beta_1 = \beta_2 = \cdots = \beta_{64} = 0$, we compare the following two nested models:

Model 1:	$y_i = \beta_0 + e_i$
Model 2:	$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_{64} x_{64i} + e_i$

Model 1 is a constrained version of Model 2 where $\beta_1 = \beta_2 = \cdots = \beta_{64} = 0$.

For each model, we can obtain the value of the likelihood, L, which is the probability of obtaining the observed data if the model were true.

The likelihood ratio test statistic is computed as $-2 \log L_1 - (-2 \log L_2)$ which under the null hypothesis H_0 follows a chi-squared distribution on q degrees of freedom, where q is the difference in the number of parameters between the two models.

The value of $-2 \log L_2$ is given at the bottom of the **Equations** window and is 10782.92. If you would like to fit Model 1, you would obtain $-2 \log L_1 = 11509.36$. The likelihood ratio statistic is $11509.36 - 10782.92 = 726.44$ which is compared to a chi-squared distribution on 64 degrees of freedom. (There are 64 more parameters in model 2 than in model 1.) A p-value for the test can be obtained as follows:

- From the **Basic Statistics** menu, select **Tail Areas**
- Under **Operation,** select **Chi-Squared**
- Next to **Value**, input the value of the test statistic, i.e, **726.44**
- Input the number of *degrees of freedom* for the test, i.e, **64**
- Click **Calculate**

You should obtain a p-value which is extremely small, suggesting that the null hypothesis $H_0 : \beta_1 = \beta_2 = \cdots = \beta_{64} = 0$ should be rejected. We conclude that there are real differences between schools in the population mean of **normexam**.

The ANOVA or fixed effects model can be used to compare any number of groups. However, there are some potential drawbacks of this approach:

[2]The test can also be extended to non-nested models – see AIC in the **Help** system.

If the sample sizes within groups are small, the estimates of the group effects may be unreliable. Also, if there are J groups to be compared, then $J-1$ parameters are required to capture group effects. If J is large, this leads to estimation of a large number of parameters.

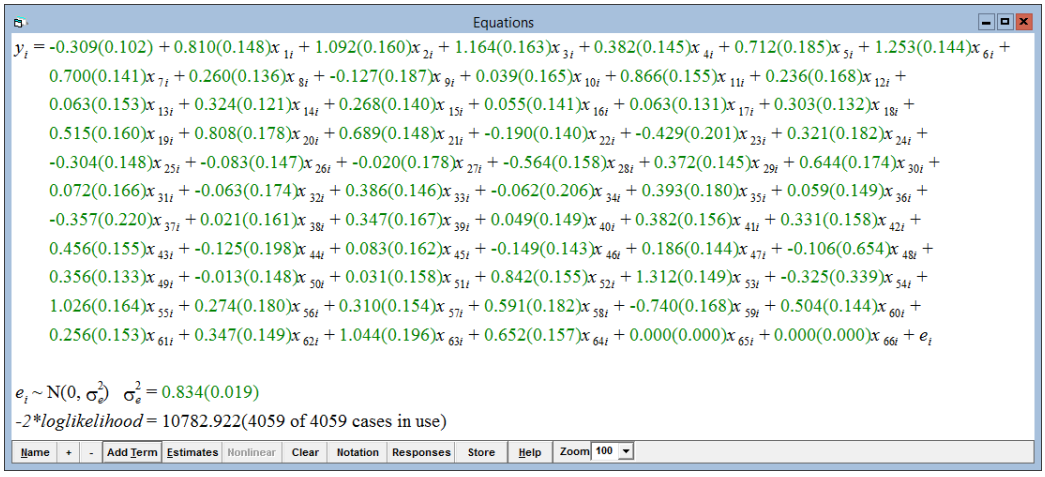
The origins of the ANOVA approach lie in experimental design where there are typically a small number of groups to be compared and all groups of interest are sampled. Often, however, we only have a sample of groups (e.g. a sample of schools) and it is the population of groups from which our sample was drawn which is of interest. The ANOVA model does not allow us to make inferences beyond the groups in our sample.

Additional explanatory variables may be added to the fixed effects model, but the effects of group-level variables cannot be separately estimated. They are confounded with the group effects. For example, if we include $x_1, x_2, \ldots, x_{64}$ in the regression model, we cannot estimate the effect of school-level characteristics such as school gender (i.e, boys' school, girls' school or mixed) on exam performance. This is because any school-level variable can be expressed as a linear combination of $x_1, x_2 \ldots, x_{64}$.

As an illustration of this last point, let's add the school-level variable, school gender (**schgend**) to the model. This variable is coded as follows: 1 for a mixed school, 2 for a boys' school and 3 for a girls' school.

- Click on the **Add Term** button
- From the drop-down list labelled **variable**, select **schgend**
- Click **Done**
- Click the **Start** button to run the model

We obtain the following results:



The first category of **schgend** (mixed schools) is taken by default as the reference for the effects of school gender, so the overall reference category

now becomes school 65 and mixed schools. The model attempts to separate school gender effects from the 64 school dummy effects. You can see that the coefficients of **boysch** and **girlsch** are estimated as 0. This tells us that this type of ANOVA or fixed effects model cannot separate out these two sets of effects because of the confounding problem described above.

2.5 Comparing means: Random effects or multilevel model

If we are interested in describing the means of a large number of groups, an alternative to the fixed effects model of equation (2.3) is a *random effects* or *multilevel* model. In a random effects model, group effects (represented by u_{0j} in equation (2.4) below) are assumed to be random, usually following a Normal distribution. The population is considered to have a two-level hierarchical structure with lowest level units at level 1, nested within groups at level 2. For example, in our educational example we have students at level 1 and schools at level 2. The residual is now partitioned into a level 1 component e_{ij} and a level 2 component u_{0j}.

The random effects model with no explanatory variables can be written as:

$$\begin{aligned} y_{ij} &= \beta_{0j} + e_{ij} \\ \beta_{0j} &= \beta_0 + u_{0j} \\ u_{0j} &\sim \mathrm{N}(0, \sigma^2_{u0}) \\ e_{ij} &\sim \mathrm{N}(0, \sigma^2_e) \end{aligned} \tag{2.4}$$

The first, second and last lines of the random effects model are equivalent to the fixed effects ANOVA model in equation (2.3), with $u_{0j} = \beta_j$. The difference is the way in which the between school differences are treated. In this model u_{0j}, the school effects, are assumed to be random variables coming from a Normal distribution with variance σ^2_{u0}. In fact, we no longer need to choose a reference category as it is more convenient to regard β_0 as the overall population mean with the u_{0j} representing each school's difference from this mean. It follows that the mean value of the random variable u_{0j} is zero. If, additionally, we assume Normality we can then describe its distribution in terms of the mean and variance as in line 3 of (2.4). This type of model is sometimes called a *variance components model*, owing to the fact that the residual variance is partitioned into components corresponding to each level in the hierarchy. The variance between groups is σ^2_{u0} and the variance between individuals within a given group is σ^2_e.

The similarity between individuals in the same group is measured by the *intra-class correlation*, where 'class' may be replaced by whatever defines

groups:

$$\frac{\sigma^2_{u0}}{\sigma^2_{u0} + \sigma^2_e}$$

The intra-class correlation measures the extent to which the y-values of individuals in the same group resemble each other as compared to those from individuals in different groups. It may also be interpreted as the proportion of the total residual variation that is due to differences between groups, and is referred to as the *variance partition coefficient* (VPC) as this is the more usual interpretation (see Goldstein (2003), pp 16-17).

Comparing a random effects model to a fixed effects model

In the multilevel approach, the groups in the sample are treated as a random sample from a population of groups. The variation between groups in this population is σ^2_{u0}. However, the number of groups should be reasonably large. If J is small, group effects can be captured using fixed effects, i.e, including dummy variables for groups as explanatory variables. Regardless of the number of groups to be compared, only one additional parameter, σ^2_{u0}, is required to capture group effects.

Categorical and continuous group-level explanatory variables may be added to the model; their effects are not confounded with u_{0j} as in the fixed effects model. An illustration of this point is given later in the chapter.

The random effects model (2.4) is specified in MLwiN as follows. First, remove the set of 64 school dummy variables from the model:

- Click on one of the school dummy variables in the **Equations** window
- Click the **Delete Term** button
- In the dialogue box that appears click **Yes** to remove the school level dummy variables

Now remove **boysch** and **girlsch**, the dummy variables for **schgend**:

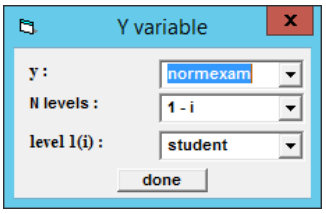
- Click on one of the two school gender dummy variables in the **Equations** window
- Click the **Delete Term** button
- In the dialogue box that appears, click **Yes** to remove the school level dummy variables

You should now have a single-level model with an intercept term, but no other explanatory variables.

The next step is to specify the two levels in the hierarchical structure. At level 1 we have students, as before, but we now have schools at level 2. In the Equations window:

- Click on **normexam** (or **y** if you are viewing mathematical symbols)

The **Y variable** window will appear:



At the moment this window shows that our y variable is **normexam**, we have specified a single-level model, and the observations are made on students. To specify a two-level hierarchical structure, with students nested within schools:

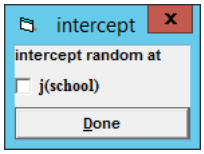
- From the **N levels:** list, select **2-ij**
- From the **level 2 (j):** list, select **school**
- Click **done**

Note that by convention MLwiN uses the suffix i for level 1 and j for level 2.

To add u_{0j} to the random part of the model, we need to specify that the intercept β_0 is random at the school level:

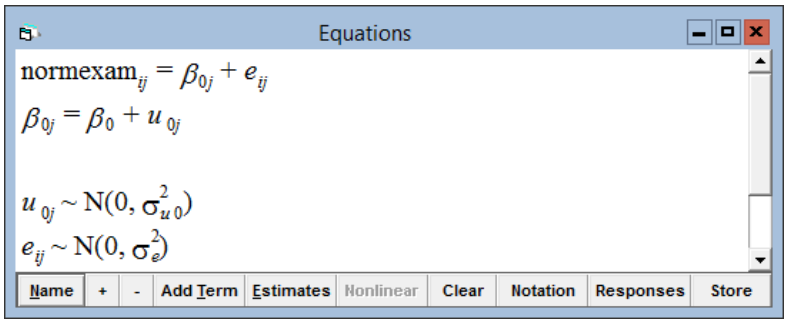
- Click β_0 in the **Equations** window (You may have to click the **Estimates** button first.)

The following window appears:



- Check the box labelled **j (school)**
- Click **Done**

This produces:


Equations
$\text{normexam}_{ij} = \beta_{0j} + e_{ij}$
$\beta_{0j} = \beta_0 + u_{0j}$
$u_{0j} \sim \text{N}(0, \sigma^2_{u0})$
$e_{ij} \sim \text{N}(0, \sigma^2_e)$
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

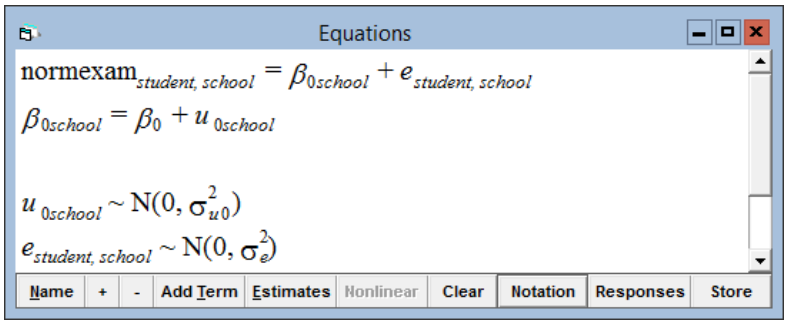
which is exactly the form of equation (2.4). You may need to click the + button a couple of times in order to see the complete model specification.

In summary, the model that we have specified allows the mean of **normexam** to vary across schools. The model allows the mean for school j to depart (be raised or lowered) randomly from the overall mean by an amount u_{0j}, which is assumed to be a Normally distributed random quantity (as stated in (2.4)). The ith student in the jth school departs from their school mean by an amount e_{ij}.

Just as we can toggle between xs and actual variable names using the **Name** button, we can also show variable names for subscripts. To change the display mode in this way:

- Click the **Notation** button
- Check the **subscripts as names** box
- Click **Done**

This produces:


Equations
$\text{normexam}_{\text{student, school}} = \beta_{0\text{school}} + e_{\text{student, school}}$
$\beta_{0\text{school}} = \beta_0 + u_{0\text{school}}$
$u_{0\text{school}} \sim \text{N}(0, \sigma^2_{u0})$
$e_{\text{student, school}} \sim \text{N}(0, \sigma^2_e)$
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

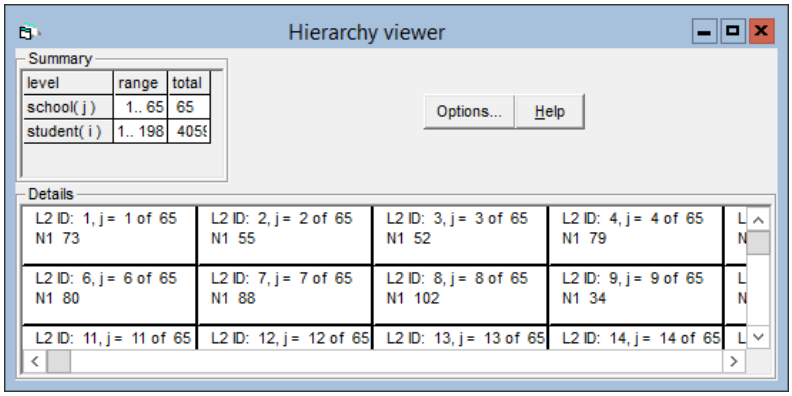
This display is somewhat verbose but a little more readable than the default subscript display. You can switch back to the default subscripts by doing the following:

- Click the **Notation** button
- Uncheck the **subscripts as names** box, and click **Done**

Before running a model it is always a good idea to get MLwiN to display a summary of the hierarchical structure to make sure that the structure MLwiN is using is correct. To do this:

- Select the **Model** menu
- Select **Hierarchy Viewer**

This produces:


Hierarchy viewer
Summary
level
range
total
school(j)
1.. 65
65
student(i)
1.. 198
Options...
Help
Details
L2 ID: 1, j = 1 of 65
N1 73
L2 ID: 2, j = 2 of 65
N1 55
L2 ID: 3, j = 3 of 65
N1 52
L2 ID: 4, j = 4 of 65
N1 79
L2 ID: 6, j = 6 of 65
N1 80
L2 ID: 7, j = 7 of 65
N1 88
L2 ID: 8, j = 8 of 65
N1 102
L2 ID: 9, j = 9 of 65
N1 34
L2 ID: 11, j = 11 of 65
L2 ID: 12, j = 12 of 65
L2 ID: 13, j = 13 of 65
L2 ID: 14, j = 14 of 65

The top **summary** grid shows, in the **total** column, that there are 4059 pupils in 65 schools. The **range** column shows that there are a maximum of 198 pupils in any school. The **Details** grid shows information on each school. 'L2 ID' means 'level 2 identifier value', so that the first cell under **Details** relates to school 1. If when you come to analyse your own data the hierarchy that is reported does not conform to what you expect, then the most likely reason is that your data are not sorted in the manner required by MLwiN.

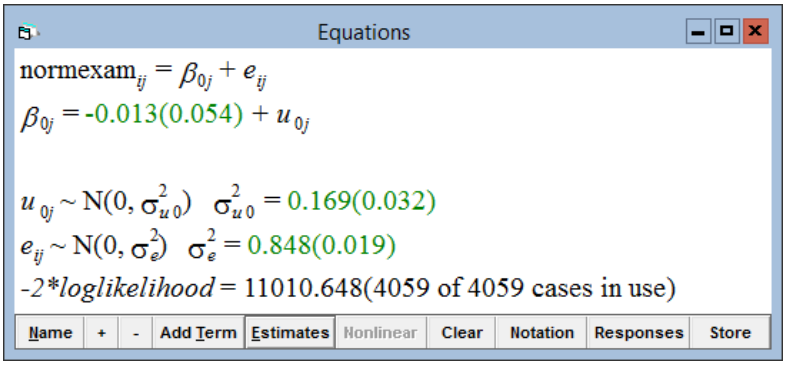
We are now ready to estimate the model. The estimation procedure for a multilevel model is iterative. The default method of estimation is iterative generalised least squares (IGLS). This is noted on the right of the **Stop** button, and it is the method we shall use. The **Estimation control** button is used to vary the method, to specify the convergence criteria, and so on. See the **Help** system for further details.

- Click **Start**

You will now see the progress gauges at the bottom of the screen fill up with green as the estimation proceeds alternately for the random and fixed

parts of the model. In the case of a single-level model, estimated using OLS, estimates are obtained after iteration 1 at which point the blue highlighted parameters in the **Equations** window change to green. For a multilevel model, however, an iterative estimation procedure is used and more iterations will be required. Estimation is completed in this case at iteration 2 using the Iterative Generalised Least Squares (IGLS) procedure. Convergence is judged to have occurred when, for each of the parameter estimates, the relative differences between two iterations is less than a given tolerance, which is $10^{-2} = 0.01$ by default but can be changed from the **Options** menu.

You should obtain the following results:


Equations
normexam$_{ij}$ = β_{0j} + e_{ij}
β_{0j} = -0.013(0.054) + u_{0j}
u_{0j} ~ N(0, σ^2_{u0}) σ^2_{u0} = 0.169(0.032)
e_{ij} ~ N(0, σ^2_e) σ^2_e = 0.848(0.019)
-2*loglikelihood = 11010.648(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The overall mean of **normexam** is estimated as $\hat{\beta}_0 = -0.013$. The means for the different schools are distributed about their overall mean with an estimated variance of 0.169. The response variable is standardised to have a Normal distribution with mean of 0 and variance of 1, which is why the estimated mean $\hat{\beta}_0$ is very close to zero and the total variance (obtained by adding the level 1 and level 2 variances) is very close to 1. The between-school variance is estimated as $\hat{\sigma}^2_u = 0.169$, and the variance between pupils within schools is estimated as $\hat{\sigma}^2_e = 0.848$.

From a Normal test of $H_0 : \sigma^2_{u0} = 0$ (analogous to testing $H_0 : \beta_1 = \beta_2 = \cdots \beta_{64} = 0$ in the fixed effects model), this variance appears to be significantly different from zero ($Z = 0.169/0.032 = 5.3$, $p < 0.001$). However, judging significance for variances (and assigning confidence intervals) is not as straightforward as for the fixed part parameters because the distribution of the estimated variance is only *approximately* Normal. The Normal test therefore provides an approximation that can act as a rough guide. A preferred test is the likelihood ratio test. In a likelihood ratio test of $H_0 : \sigma^2_{u0} = 0$, we compare the model above with a model where σ^2_{u0} is constrained to equal zero, i.e, the single-level model with only an intercept term. The value of the likelihood ratio statistic, obtained from the two models' loglikelihoods, is $11509.36 - 11010.65 = 498.71$, which is compared to a chi-squared distribution on 1 degree of freedom. We conclude that there is significant variation between schools.

Note that the likelihood ratio statistic is very different from $Z^2 = 28.09$ which also has a chi squared distribution on 1 degree of freedom, so the Normal test based on Z is a poor approximation in this case.

The variance partition coefficient is:

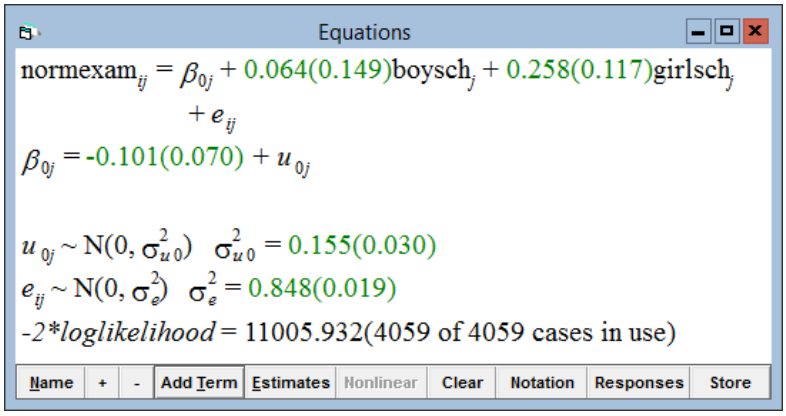
$$\frac{0.169}{0.169 + 0.848} = 0.166$$

About 17% of the total variance in **normexam** may be attributed to differences between schools.

Unlike the fixed ANOVA model, we can now investigate the effects of school-level variables. For example, we can add **schgend**:

- Click on the **Add Term** button
- From the drop-down list labelled **variable** select **schgend**
- Click **done**
- Click the **Start** button to run the model

You should then see the following results:



*Note that MLwiN recognises that **schgend** is a school level variable (since all students in the same school have the same value of **schgend**) and therefore assigns a single subscript, j, to the dummy variables **boysch** and **girlsch**.*

The reference category for **schgend** is mixed schools, so β_0 is the mean for children attending a mixed school, estimated by -0.101. β_1 is the coefficient of **boysch** and represents the mean difference in exam scores between children in a boys' school and children in a mixed school. Likewise β_2 is the coefficient of **girlsch** and it represents the mean difference in exam scores between children in a girls' school and children in a mixed school. We see

that children from girls' schools fare better by 0.258 of a standard deviation unit, on average, in their exam scores than children from a mixed school. If we compare the between-school variation for this model and the previous model, which excluded school gender terms, we see a reduction from 0.169 to 0.155. This reduction indicates that very little of the differences between schools is explained by school gender. The ability to estimate between-group variation and also include group-level covariates in an attempt to explain between-group variation is a great strength of multilevel modelling.

- Please save your worksheet at this point by clicking on the **File** menu, and selecting **Save worksheet As...**
- Type **tutorial2** in the box next to **Filename:**
- Click **Save**

Chapter learning outcomes

- ⋆ How to fit a single-level (fixed effects) regression model to compare the means of two or more groups
- ⋆ What a multilevel, variance components model is
- ⋆ The difference between fixed effects and random effects models
- ⋆ The equations used to describe these models
- ⋆ Switching between different display modes in MLwiN, using the **Notation**, **Name**, +, − and **Estimates** buttons
- ⋆ How to construct, estimate and interpret these models using the **Equations** window in MLwiN
- ⋆ How to carry out Normal tests and likelihood ratio tests of significance

Chapter 3

Residuals

3.1 What are multilevel residuals?

Towards the end of the last chapter, we fitted a multilevel model that allowed for school effects on exam scores at age 16, **normexam**. The model was given in equation (2.4):

$$\begin{aligned} y_{ij} &= \beta_{0j} + e_{ij} \\ \beta_{0j} &= \beta_0 + u_{0j} \\ u_{0j} &\sim \mathrm{N}(0, \sigma^2_{u0}) \\ e_{ij} &\sim \mathrm{N}(0, \sigma^2_e) \end{aligned} \tag{2.4}$$

The u_{0j} terms are the school random effects, sometimes referred to as school residuals. In a fixed effects (ANOVA) model the school effects, represented in equation (2.3) by $\beta_0 + \beta_j$, are treated as fixed parameters for which direct estimates are obtained. In a multilevel (random effects) model, the school effects are random variables whose distribution is summarised by two parameters, the mean (zero) and variance σ^2_{u0}. However, if we wish to make comparisons between schools we need to estimate the individual school residuals in some way, after having fitted the model. In this chapter, we describe how school residuals can be estimated, how these estimates can be obtained using MLwiN, and how the estimated residuals can be used for checking model assumptions. We conclude by discussing interpretation.

In this chapter we will work with the multilevel model specified in equations (2.4) above. Chapters 2 to 7 consist of a series of multilevel analyses on the tutorial data set. In order to make each of these chapters self-contained, we always start by opening the file **tutorial.ws**:

- Select the **Open Worksheet** option on the **File** menu
- Open the file **tutorial.ws**
- Select the **Equations** menu item from the **Model** menu
- Click the **Notation** button
- In the **Notation** window, clear the box beside **general**
- Click **Done**

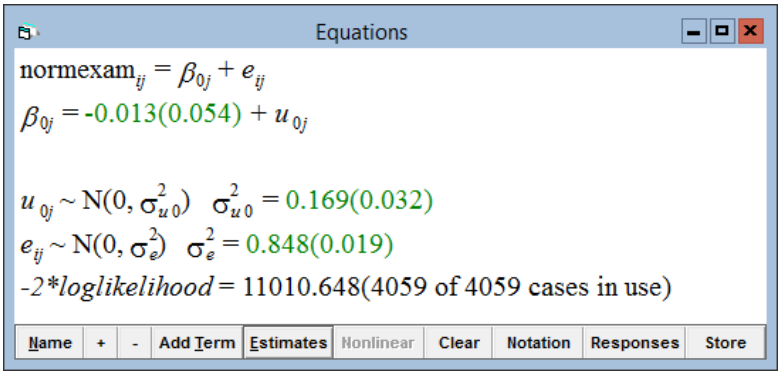
To demonstrate calculation of residuals in MLwiN, let's fit model (2.4):

- Click on y in the **Equations** window
- In the **Y variable** window, select **y: normexam**
- Select **N levels: 2-ij**
- Select **level 2(j): school**
- Select **level 1(i): student**
- Click **done**
- In the **Equations** window, click on β_0
- In the **intercept** window, check **school(j)**
- Click **Done**

Now run the model and view the estimates:

- Click **Start** on the main toolbar
- Click **Name** then **Estimates** twice in the **Equations** window

This produces:



The current model is a two-level variance components model, with the overall mean of the dependent variable **normexam** defined by a fixed coefficient β_0. The second level was added by allowing the mean for the jth school to be

raised or lowered from the overall mean by an amount u_{0j}. These departures from the overall mean are known as the level 2 residuals. Their mean is zero and their estimated variance of 0.169 is shown in the **Equations** window. With educational data of the kind we are analysing, they might be called the school effects. In other data sets, the level 2 residuals might be hospital, household or area effects, etc.

The true values of the level 2 residuals are unknown, but we will often need to obtain estimates of them. We might reasonably ask for the effect on student attainment of one particular school. We can in fact predict the values of the residuals, given the observed data and the estimated parameters of the model (see Goldstein (2003), Appendix 2.2). In OLS multiple regression, we can estimate the residuals simply by subtracting the individual predictions from the observed values. In multilevel models with residuals at each of several levels, a more complex procedure is needed.

Suppose that y_{ij} is the observed value for the ith student in the jth school and that $\hat{y}_{ij}$ is the predicted value from the regression, which for the current model will equal the overall mean of **normexam**. Then the *raw residual* for this subject is $r_{ij} = y_{ij} - \hat{y}_{ij}$. The raw residual for the jth school is the mean of the r_{ij} for the students in the school. Write this as r_{+j}. Then the estimated level 2 residual for this school is obtained by multiplying r_{+j} by a factor as follows:

$$\hat{u}_{0j} = \frac{\sigma^2_{u0}}{\sigma^2_{u0} + \sigma^2_e/n_j} r_{+j}$$

where n_j is the number of students in school j.

The multiplier in the above formula is always less than or equal to 1 so that the estimated residual is usually less in magnitude than the raw residual. We say that the raw residual has been multiplied by a *shrinkage factor* and the estimated residual is sometimes called a *shrunken residual*. The shrinkage factor will be noticeably less than 1 when σ^2_e is large compared to σ^2_{u0} or when n_j is small (or both). In either case we have relatively little information about the school (its students are very variable or few in number) and the raw residual is pulled in towards zero. In future we shall use the term 'residual' to refer to a shrunken residual.

To explain further the idea of shrinkage, consider the case where we have no observations at all in a particular school, perhaps one that was not in our sample of schools. Our best estimate of that school's performance is the overall mean $\hat{\beta}_0$. For a school with observations on a small number of students, we can provide a more individualised estimate of its mean performance, but we know that the mean is estimated imprecisely. Rather than trusting this imprecise mean on its own, we may prefer to use the information that the school is a member of the population of schools whose parameters we have

estimated from the data, by using a weighted combination of the estimated school mean and the estimated population mean. This implies shrinking the observed school mean towards the centre of the population, and the optimal factor for this purpose is the shrinkage factor given above.

It is the shrinkage factor that causes the difference between the ANOVA group means and group means estimated from a multilevel analysis. For example, suppose the school with the highest actual mean contains only two pupils. ANOVA would just reproduce the actual mean from the sample data, giving it a large standard error. The mean for this school based on a multilevel model will be shrunk in towards the overall mean for all schools.

Note that having estimated the level 2 residuals we can estimate the level 1 residuals simply by the formula

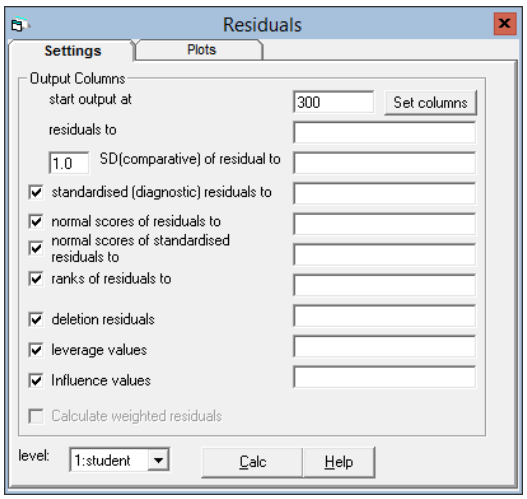
$$\hat{e}_{ij} = r_{ij} - \hat{u}_{0j}$$

MLwiN is capable of calculating residuals at any level and of providing standard errors for them. These can be used for comparing higher-level units (such as schools) and for model checking and diagnosis.

3.2 Calculating residuals in MLwiN

We can use the **Residuals** window in MLwiN to calculate residuals. Let's take a look at the level 2 residuals in our model:

- Select the **Model** menu
- Select **Residuals**
- Select the **Settings** tab of the **Residuals** window



The *comparative standard deviation* (SD) of the residual is defined as the standard deviation of $\hat{u}_{0j} - u_{0j}$ and is used for making inferences about the unknown underlying value u_{0j}, given the estimate $\hat{u}_{0j}$. The *standardised residual* is defined as $\hat{u}_{0j}/SD(\hat{u}_{0j})$ and is used for diagnostic plotting to ascertain Normality etc.

As you will see, this window permits the calculation of the residuals and of several functions of them. We need level 2 residuals, so at the bottom of the window:

- From the **level:** drop-down list, select **2:school**

You also need to specify the columns into which the computed values of the functions will be placed:

- Click the **Set columns** button

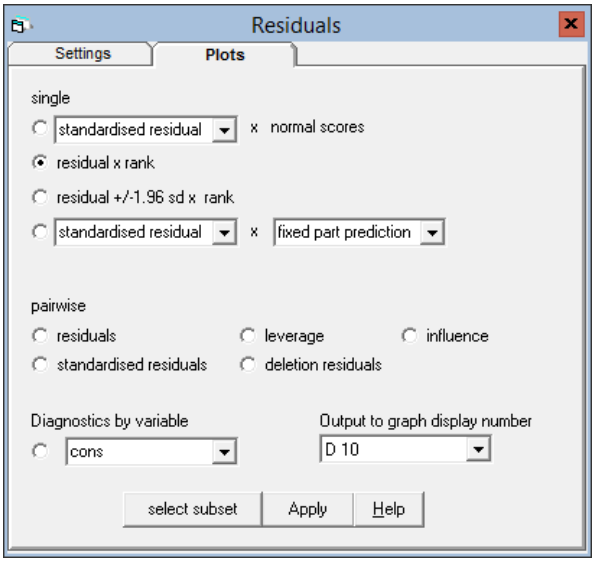
The nine boxes beneath this button are now filled in grey with column numbers running sequentially from C300. These columns are suitable for our purposes, but you can change the starting column by editing the **start output at** box. You can also change the multiplier applied to the standard deviations; by default 1× SD will be stored in **c301**:

- Edit the SD multiplier to be **1.96**
- Click **Calc** (to calculate columns **c300** to **c308**)

Having calculated the school residuals, we need to inspect them. MLwiN provides a variety of graphical displays for this purpose. The most useful of these are available directly from the **Residuals** window:

- Click on the **Plots** tab

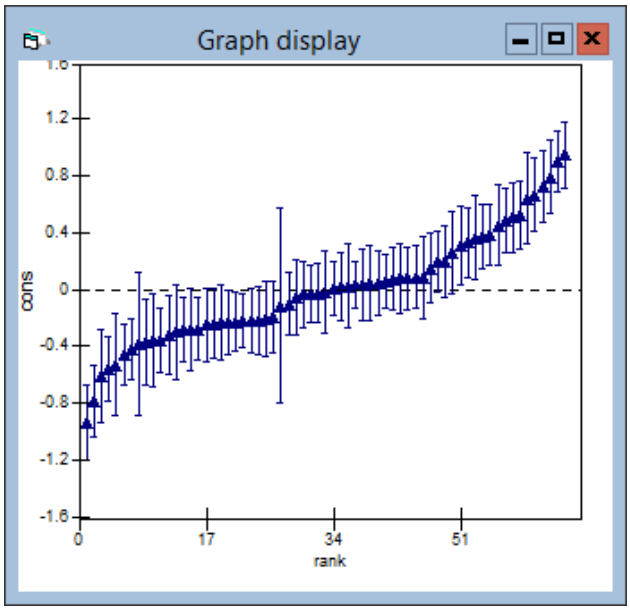
This brings up the following window:



One useful display plots the residuals in ascending order with their 95% confidence limit. To obtain this:

- Click on the third option in the **single** frame (residual + / - 1.96 SD x rank)
- Click **Apply**

The following graph appears:



This is sometimes known (for obvious reasons) as a caterpillar plot. We have 65 level 2 residuals plotted, one for each school in the data set. Looking at the confidence intervals around them, we can see a group of about 20 schools at the lower and upper end of the plot where the confidence intervals for their residuals do not overlap zero. Remembering that these residuals represent school departures from the overall average predicted by the fixed parameter

β_0, this means that these are the schools that differ significantly from the average at the 5% level.

See Goldstein & Healy (1995) for further discussion on how to interpret and modify such plots when multiple comparisons among level 2 units are to be made. Comparisons such as these, especially of schools or hospitals, raise difficult issues: in many applications, such as here, there are large standard errors attached to the estimates. Goldstein & Speigelhalter (1996) discuss this and related issues in detail.

Note: You may find that you sometimes need to resize graphs in MLwiN to obtain a clear labelling of axes.

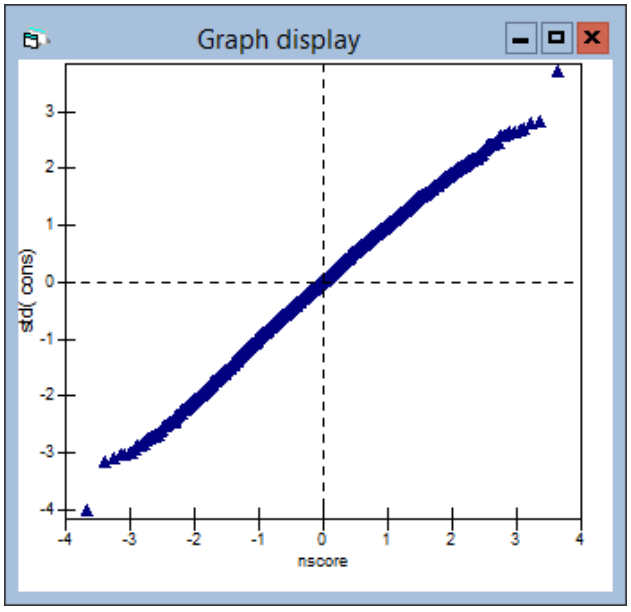
3.3 Normal plots

So far, we have looked at using estimated level 2 residuals for interpretation purposes. For example, when the level 2 units are schools, the level 2 residuals can be interpreted as school effects. Estimated residuals, at any level, can also be used to check model assumptions. One such assumption is that the residuals at each level follow Normal distributions. This assumption may be checked using a Normal probability plot, in which the ranked residuals are plotted against corresponding points on a Normal distribution curve. If the Normality assumption is valid, the points on a Normal plot should lie approximately on a straight line.

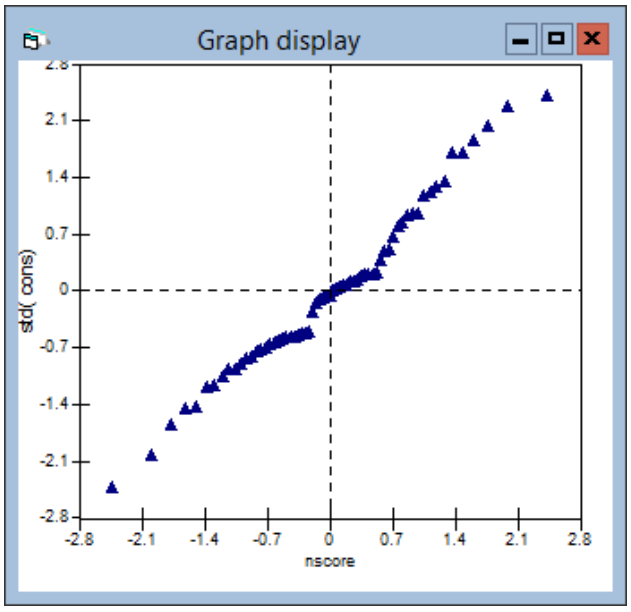
We will begin by examining a Normal plot of the level 1 residuals. To produce a Normal plot in MLwiN:

- Select the **Model** menu
- Select **Residuals**
- Click on the **Settings** tab of the **Residuals** window
- From the **level:** list select **1:student**
- Click the **Set columns** button
- Click **Calc**
- Click on the **Plots** tab
- Click on the first option, **standardised residual x normal scores**
- Click **Apply**

You will obtain the following plot. The plot looks fairly linear, which suggests that the assumption of Normality is reasonable. This is not surprising in this case since our response variable has been normalised.



To produce a Normal plot of the level 2 residuals, just repeat the steps described above, but next to **level:** in the **Settings** tab replace **1:student** by **2:school** to calculate the school residuals. You should obtain the following plot, which again looks fairly linear.



Please save your worksheet at this point:

- Click on the **File** menu and select **Save worksheet as ...**
- Type **tutorial3** in the box next to **Filename:**
- Click **Save**

Chapter learning outcomes

- ⋆ Multilevel residuals are shrunken towards zero, and shrinkage increases as n_j decreases
- ⋆ How to calculate residuals in MLwiN
- ⋆ How to produce and interpret Normal probability plots

Chapter 4

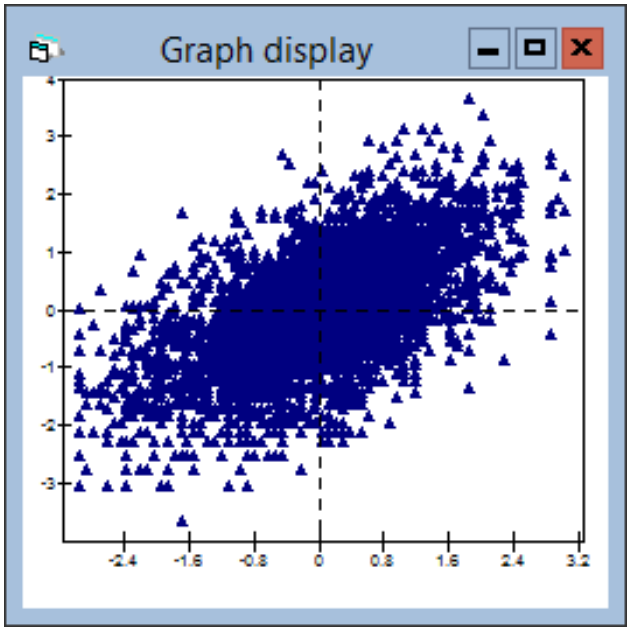
Random Intercept and Random Slope Models

4.1 Random intercept models

In any serious study of school effects we need to take into account student intake achievements in order to make 'value-added' comparisons between schools. In this chapter, we consider whether the differences in **normexam** between schools remain after adjusting for a measure of achievement on entry to secondary school, **standlrt**. **standlrt** is the students' score at age 11 on the London reading test, standardised to produce z-scores. We also consider random effects models, which allow the effect of intake score to vary across schools. Let's start by plotting the response, **normexam**, against **standlrt**:

- Open the **tutorial.ws** worksheet
- Select **Customised Graph(s)** from the **Graphs** menu
- In the drop-down list labelled **y,** select **normexam**
- In the drop-down list labelled **x,** select **standlrt**
- Click on the **Apply** button

The following graph will appear:



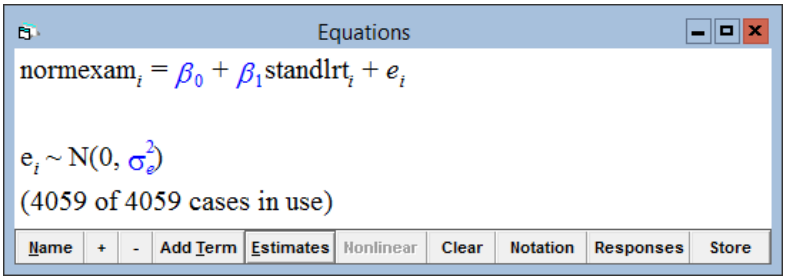
The plot shows, as might be expected, a positive correlation where pupils with higher intake scores tend to have higher outcome scores. We can fit a simple linear regression to this relationship:

- Select the **Equations** menu item from the **Model** menu
- Click the **Notation** button
- In the **Notation** window, clear the box beside **general**
- Click **Done**

Now set up the model:

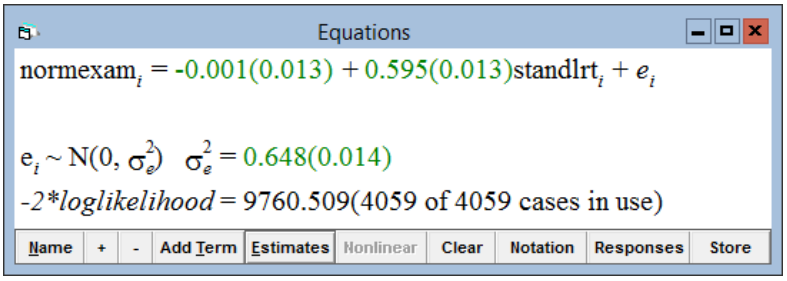
- Click on **y** in the **Equations** window
- In the **Y variable** window, select **y**: **normexam**
- Select **N levels: 1-i**
- Select **level 1(i): student**
- Click **Done**
- In the **Equations** window, click on the **Add term** button
- From the **Specify term** window's variable drop-down list, select **standlrt**
- Click **Done**
- Click the **Estimates** button

The Equations window now looks like this:


Equations
normexam$_i$ = β_0 + $\beta_1$standlrt$_i$ + e_i
$e_i \sim N(0, \sigma_e^2)$
(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

Estimate the model and view the results:

- Click the **Start** button on the main toolbar
- Click the **Estimates** button on the **Equations** window


Equations
normexam$_i$ = -0.001(0.013) + 0.595(0.013)standlrt$_i$ + e_i
$e_i \sim N(0, \sigma_e^2)$ σ_e^2 = 0.648(0.014)
-2*loglikelihood = 9760.509(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The Equations window above shows a simple linear regression model that describes the positive relationship between **normexam** and **standlrt**. The equation of the estimated regression line is:

$$\textbf{normexam} = -0.001 + (0.595 \times \textbf{standlrt})$$

An increase of 1 unit on the intake **standlrt** variable increases the expected outcome examination score, **normexam**, by 0.60 units. The variability of the students' scores around the overall average line is 0.65.

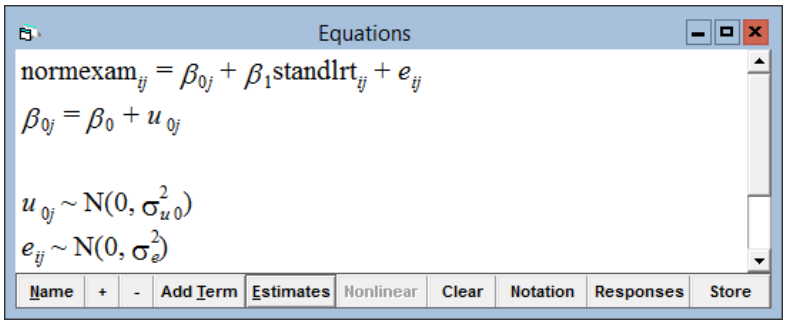
We can now address the question of whether schools vary after having taken account of intake using either a fixed or random effects model. We shall work with the random effects multilevel model:

- Click on **normexam** in the **Equations** window
- In the **N levels** list, select **2-ij**
- In the **level 2(j)** list, select **school**
- Click **Done**

We want to allow the intercept term to vary randomly across schools. To do this:

- Press the **Estimates** button to display mathematical symbols
- Click on the β_0 term in the **Equations** window (actually the -0.001)
- Check the box labelled **j(school)**
- Click **Done**

The Equations window will show the updated model structure:

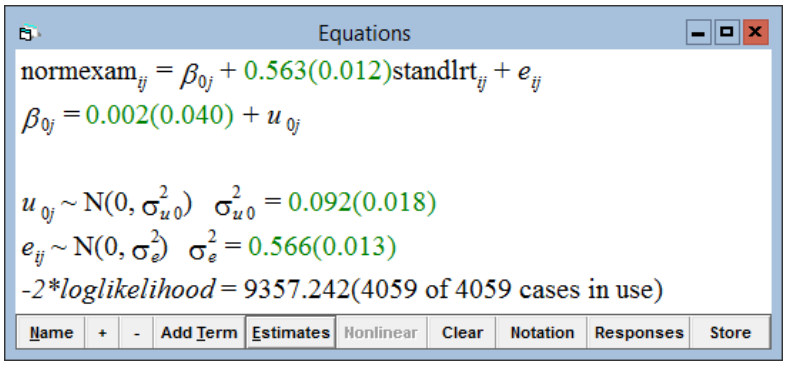

Equations
normexam$_{ij}$ = β_{0j} + $\beta_1$standlrt$_{ij}$ + e_{ij}
β_{0j} = β_0 + u_{0j}
u_{0j} ~ N(0, σ^2_{u0})
e_{ij} ~ N(0, σ^2_e)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

This is an extended version of the random effects, multilevel, model in equation (2.4) that we used in Chapter 2 to estimate between school variability, where we have now taken some account of pupil intake ability by including a term for **standlrt** in our model. Every school now has its own intercept, β_{0j}, but all schools share a common slope. This amounts to fitting a series of parallel lines, one for each school.

Let's now run the model and view the estimates:

- Click the **Start** button on the main window's toolbar
- Click the **Estimates** button (twice if necessary) in the **Equations** window

This produces the following:


Equations
normexam$_{ij}$ = β_{0j} + 0.563(0.012)standlrt$_{ij}$ + e_{ij}
β_{0j} = 0.002(0.040) + u_{0j}
u_{0j} ~ N(0, σ^2_{u0}) σ^2_{u0} = 0.092(0.018)
e_{ij} ~ N(0, σ^2_e) σ^2_e = 0.566(0.013)
-2*loglikelihood = 9357.242(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

Recall that our model amounts to fitting a set of parallel straight lines to the data from the different schools. The slopes of the lines are all the same, and the fitted value of the common slope is 0.563 with a standard error of 0.012

(clearly, this is highly significant). However, the intercepts of the lines vary. Their mean is 0.002 and this has a standard error of 0.040. Not surprisingly with Normalised and standardised data, the mean intercept is close to zero. The intercepts for the different schools are the level 2 residuals u_{0j} and these are distributed around zero with a variance shown on line 3 of the display as 0.092 (standard error 0.018). Of course, the actual data points do not lie exactly on the straight lines; they vary about them with amounts given by the level 1 residuals e_{ij}, and these have a variance estimated as 0.566 (standard error 0.013). We saw in the previous chapter chapter how MLwiN enables us to estimate and plot the residuals and we shall use this further in the next chapter where we will see how we can look at residual and other plots together in order to obtain a better understanding of the model.

The likelihood ratio test comparing the single level linear regression model with the multilevel model, where we estimate the between school variation in the intercepts, is $9760.5 - 9357.2 = 403.3$ with 1 degree of freedom (corresponding to the added parameter σ^2_{u0}). We therefore conclude that there is significant variability between schools even after adjusting for the students' intake achievement.

4.2 Graphing predicted school lines from a random intercept model

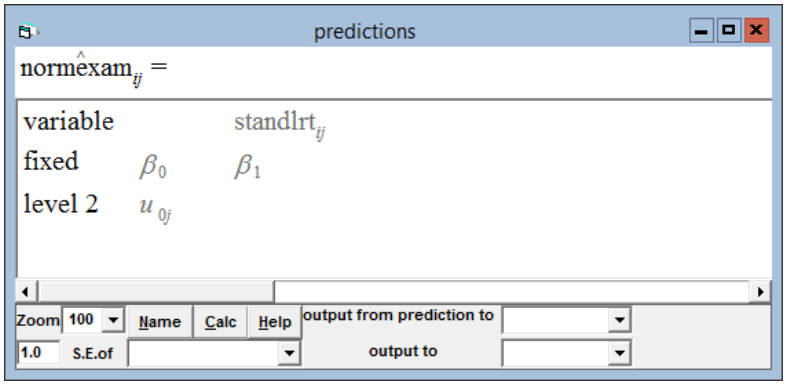
We have now constructed and fitted a *variance components* model in which schools vary only in their intercepts. It is a model of *simple* variation at level 2, which gives rise to parallel lines as illustrated in Figure 1.2 (for four schools).

To demonstrate how the model parameters we have just estimated combine to produce such parallel lines, we now introduce the **Predictions** window. This window can be used to calculate predictions from the model which can be used in conjunction with the **Customised graphs** window to graph our predicted values.

Let's start by calculating the average predicted line produced from the fixed part intercept and slope coefficients (β_0, β_1):

- Select **Predictions** from the **Model** menu

This brings up the **predictions** window:

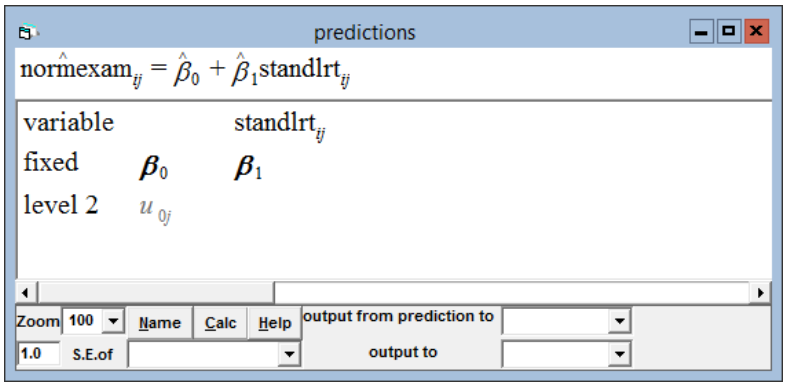


The elements of the model are arranged in two columns. Initially these columns are 'greyed out'. You build up a prediction equation in the top section of the window by selecting the elements you want from the lower section. Clicking on an element includes it in the prediction equation, and clicking again on that element removes it from the equation.

Select suitable elements to produce the desired equation, for example:

- Click on β_0 and β_1

The **predictions** window should now look like this:



The only estimates used in this equation are $\hat{\beta}_0$ and $\hat{\beta}_1$, the fixed parameters. No random quantities have been included.

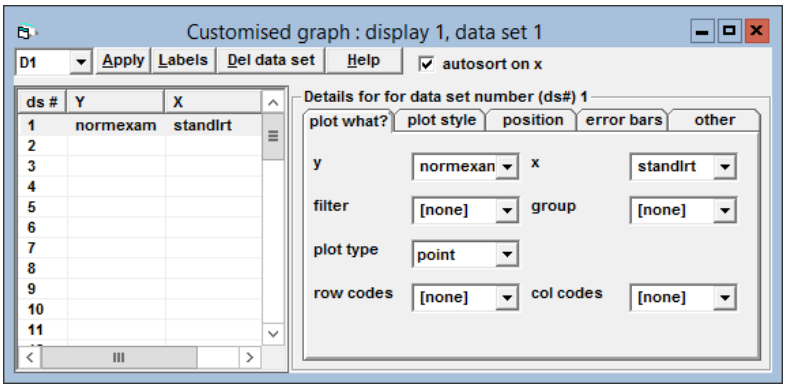
We need to specify where the output from the prediction is to go and then execute the prediction:

- In the **output from prediction to** drop-down list, select **c11**
- Click **Calc**

We now want to graph the predictions in column 11 against our predictor variable **standlrt**. We can do this using the **Customised graph(s)** window:

- Select the **Graphs** menu
- Select **Customised Graph(s)**

This produces the following window:



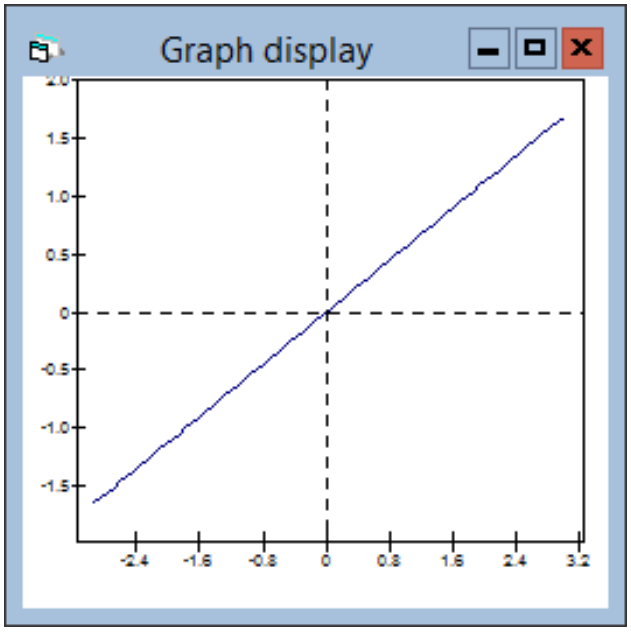
Display **D1** contains the scatter plot we specified at the start of the chapter. We will graph the prediction we have just created in a new display:

- Select **D2** (display 2) from the display drop-down list located at the top left of the **Customised graph** window

This general purpose graphing window has a great deal of functionality, described in more detail both in the help system and in the next chapter of this guide. For the moment we will confine ourselves to its more basic functions. To plot the predicted values:

- In the drop-down list labelled **y** in the **plot what?** tab, select **c11**
- In the neighbouring drop-down list labelled **x** select **standlrt**
- In the drop-down list labelled **plot type** select **line**
- Click the **Apply** button

The following graph will appear:



The prediction equation is

$$\widehat{\mathbf{normexam}}_{ij} = \hat{\beta}_0 + \hat{\beta}_1 \mathbf{standlrt}_{ij}$$

where a hat over a term means "estimate of". So substituting the estimates of the intercept and slope we get the following prediction equation.

$$\widehat{\mathbf{normexam}}_{ij} = 0.002 + 0.563\,\mathbf{standlrt}_{ij}$$

The line for the jth school departs from the average prediction line by an amount u_{0j}. The school level residual u_{0j} modifies the intercept term, but the slope coefficient β_1 is fixed. Thus all the predicted lines for all 65 schools must be parallel. The prediction equation for the jth school is therefore

$$\widehat{\mathbf{normexam}}_{ij} = (0.002 + \hat{u}_{0j}) + 0.563\,\mathbf{standlrt}_{ij} \tag{4.1}$$

We saw in the previous chapter how to get MLwiN to calculate residuals. Let's have a look at the school level residuals:

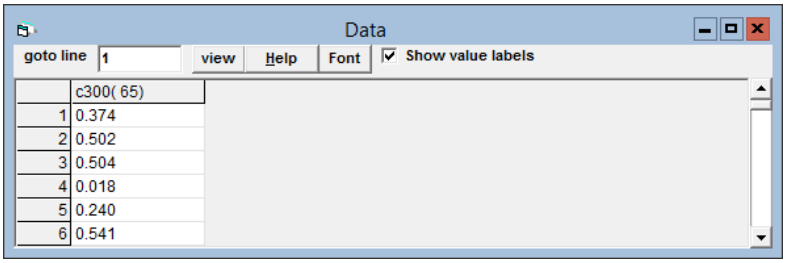
- Select **Residuals** from the **Model** menu
- Set the **level drop** down list at the bottom left of the **Residuals** window (on the **Settings** tab) to **2:school**
- Click the **Calc** button

The level 2 residuals have been written to column 300 (c300) of the worksheet, as indicated towards the top of the **output columns** section of the residuals window. We can view the data in this column by doing the following:

- Select **View or edit data** from the **Data Manipulation** menu

- Click the **view** button at the top of the window that appears
- From the drop-down list that appears, scroll to **c300**
- Click the **OK** button

This gives:



We see that column 300 contains 65 entries, one for each school. (Column lengths are displayed in brackets after column names at the top of the data grid). The intercept residual for school 1 is 0.37 and for school 2 is 0.50 and so on. We can now substitute the estimate for the jth school's residual, $\hat{u}_{0j}$, into equation 4.1 to give the prediction line for the jth school. For example, if we substitute residuals for schools 1 and 2 we get the following pair of prediction lines:

$$\widehat{\mathbf{normexam}}_{ij} = (0.002 + 0.37) + 0.563\mathbf{standlrt}_{ij} \tag{4.2}$$

$$\widehat{\mathbf{normexam}}_{ij} = (0.002 + 0.50) + 0.563\mathbf{standlrt}_{ij} \tag{4.3}$$

We can calculate and graph the prediction lines for all 65 schools to get a picture of the between-school variability using the **predictions** and **Customised graph** windows.

First we calculate the prediction lines using the **predictions** window:

- Select the **predictions** window

The predictions window is currently showing the prediction for the average line, given by the equation:

$$\widehat{\mathbf{normexam}}_{ij} = \hat{\beta}_0 + \hat{\beta}_1\mathbf{standlrt}_{ij} \tag{4.4}$$

To include the estimated school level intercept residuals in the prediction function:

- Click on the term u_{0j}

The prediction equation in the upper panel of the predictions window now becomes

$$\widehat{\mathbf{normexam}}_{ij} = \hat{\beta}_{0j} + \hat{\beta}_1 \mathbf{standlrt}_{ij} \tag{4.5}$$

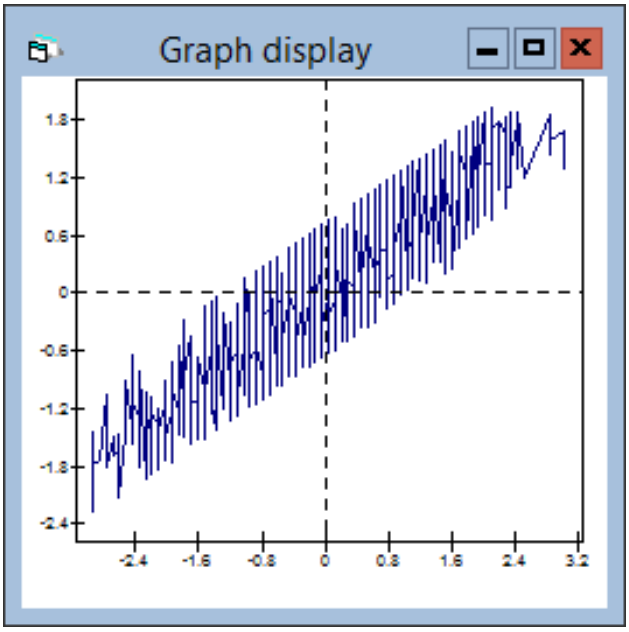
The difference between equations (4.4) and (4.5) is that the estimate of the intercept $\hat{\beta}_0$ now has a j subscript. This subscript indicates that instead of having a single intercept, we have an intercept for each school, which is formed by taking the fixed estimate and adding the estimated residual for school j, i.e, $\hat{\beta}_{0j} = \hat{\beta}_0 + \hat{u}_{0j}$.

We therefore have a regression equation for each school which when applied to the data produces 65 parallel lines. To overwrite the previous prediction in column 11 with the parallel lines:

- Click the **Calc** button in the **predictions** window

This calculation applies prediction equation (4.5) to every point in the data set. This results in a column 11 containing 4059 predicted values. The first school contains 73 students; the prediction equation for the first school becomes equation (4.2), after substituting the intercept residual for school 1 into (4.5). Equation (4.2) is applied to the first 73 values of **standlrt** resulting in the first 73 predicted points in column 11. These predicted points when plotted against the 73 **standlrt** values will lie on a straight line since prediction equation (4.2) is the equation of a straight line. The second school contains 55 students and the prediction equation for the second school (4.3) is applied to these data points resulting in predicted points 74...128 in column 11. This process is repeated for each school in the data set resulting in column 11 being filled with 4059 predicted points.

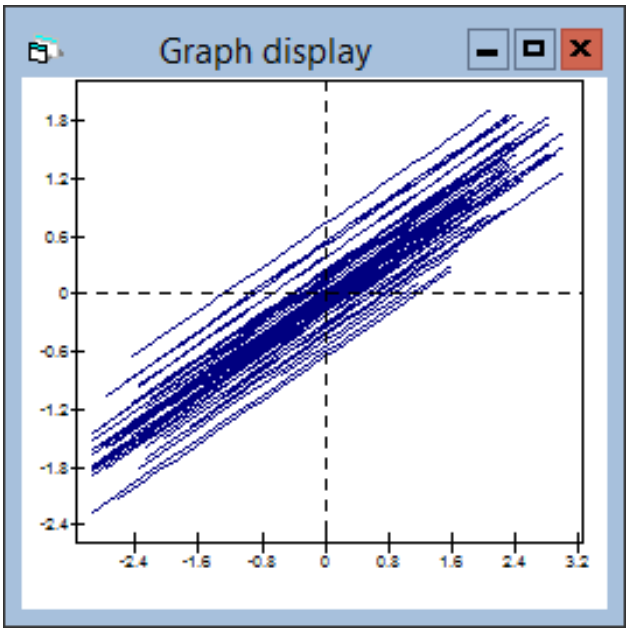
The graph display is updated automatically when column 11 is overwritten with the new prediction. However, we do not see the expected 65 lines, what we see is:



This is a plot of the predictions in column 11 against **standlrt**. The graph does not recognise that the data is grouped into 65 schools. What we need is a grouped plot:

- Select the **Customised graphs** window
- In the **group** drop-down list select **school**
- Click **Apply**

The graph display now shows the expected 65 parallel lines:



A natural next step is to construct models that allow the slopes to be different in different schools. However, before we do that we will look at another

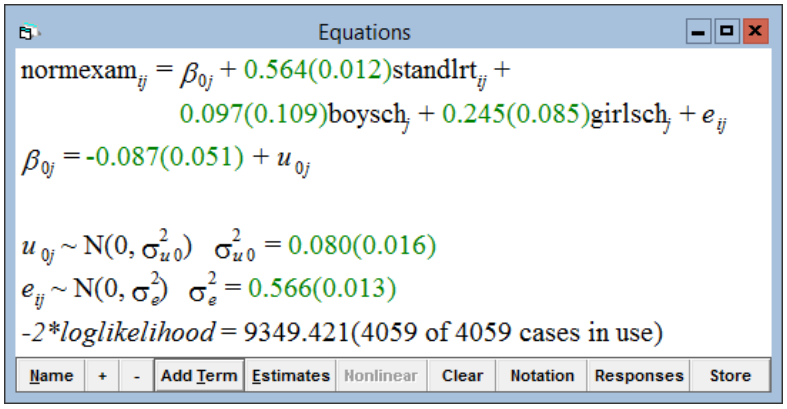
important feature of multilevel models.

4.3 The effect of clustering on the standard errors of coefficients

As was pointed out in Chapter 1, ignoring the fact that pupils are grouped within schools can cause underestimation of the standard errors of regression coefficients. This clustering can lead to incorrect inferences; since the standard errors are too small, we may infer that relationships exist between variables when they do not. To illustrate this, let's introduce the **schgend** variable again:

- Click on the **Add Term** button on the **Equations** window toolbar
- From the **Specify term** window's **variable** drop-down list, select **schgend**
- Click **Done**
- Click the **More** button to estimate the updated model

This is the model we fitted at the end of Chapter 2, except we have also included a term for **standlrt**.

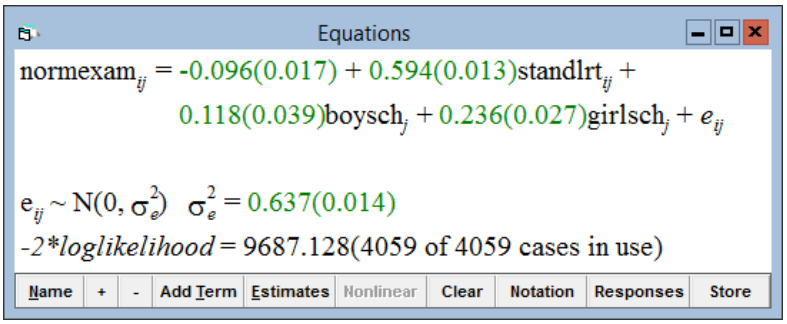

Equations
normexam$_{ij}$ = β_{0j} + 0.564(0.012)standlrt$_{ij}$ +
0.097(0.109)boysch$_j$ + 0.245(0.085)girlsch$_j$ + e_{ij}
β_{0j} = -0.087(0.051) + u_{0j}
$u_{0j} \sim N(0, \sigma^2_{u0})$ σ^2_{u0} = 0.080(0.016)
$e_{ij} \sim N(0, \sigma^2_e)$ σ^2_e = 0.566(0.013)
-2*loglikelihood = 9349.421(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

As we saw previously, both boys' and girls' schools have a higher mean **normexam** than mixed schools. To test whether boys' and girls' schools differ from mixed schools, we compare the coefficients β_2 and β_3 to their standard errors. We see that the mean for girls' schools is significantly different from the mean for mixed schools, whereas boys' schools do not differ significantly from mixed schools. Now let's fit the single-level model:

- Click on β_{0j}
- On the screen that appears uncheck the box labelled **j(school)**

- Click **Done**
- Run the model by pressing the **More** button

We then obtain the following results:


Equations
normexam$_{ij}$ = -0.096(0.017) + 0.594(0.013)standlrt$_{ij}$ +
0.118(0.039)boysch$_j$ + 0.236(0.027)girlsch$_j$ + e_{ij}
$e_{ij} \sim N(0, \sigma_e^2)$ $\sigma_e^2 = 0.637(0.014)$
-2*loglikelihood = 9687.128(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The estimated coefficients obtained from the two models are very similar. However, the standard errors are all different. The boys' school coefficient in the multilevel model is less than its standard error, and therefore not statistically significant. In the single level model, the same coefficient is three times its standard error. In this case, we would make an incorrect inference about the effect of boys' schools on achievement if we used the results from a single level model. Apart from the standard error of **standlrt**, the standard errors of the coefficients are substantially reduced in the single level model. This demonstrates why OLS regression should not be used when there are level 2 explanatory variables. Generally, the standard errors of level 1 fixed coefficients will also be underestimated in single level models.

4.4 Does the coefficient of standlrt vary across schools? Introducing a random slope

The *variance components* model that we have just worked with assumes that the only variation between schools is in their intercepts. We should allow for the possibility that the school lines have different slopes as in Figure 1.3 (in Chapter 1). This implies that the coefficient of **standlrt** will vary from school to school. Again, we can achieve this by fitting a random effects or a fixed effects model. As in Chapter 2, we would include 64 dummy variables (taking one school as the reference category) to obtain a separate intercept for each school. To allow the effect of **standlrt** to vary across schools, we would need to include a set of interaction terms, created by multiplying **standlrt** with each of the 64 school dummy variables. Taking school 65 as

the reference category, we obtain the following model:

$$\begin{aligned} y_i = \beta_0 + \beta_1 &\textbf{standlrt} \\ &+ \beta_2 \text{school\_1}_i + \ldots + \beta_{65} \text{school\_64}_i \\ &+ \beta_{66} \text{school\_1} \cdot \textbf{standlrt}_i + \cdots + \beta_{129} \text{school\_64} \cdot \textbf{standlrt}_i + e_i \end{aligned}$$

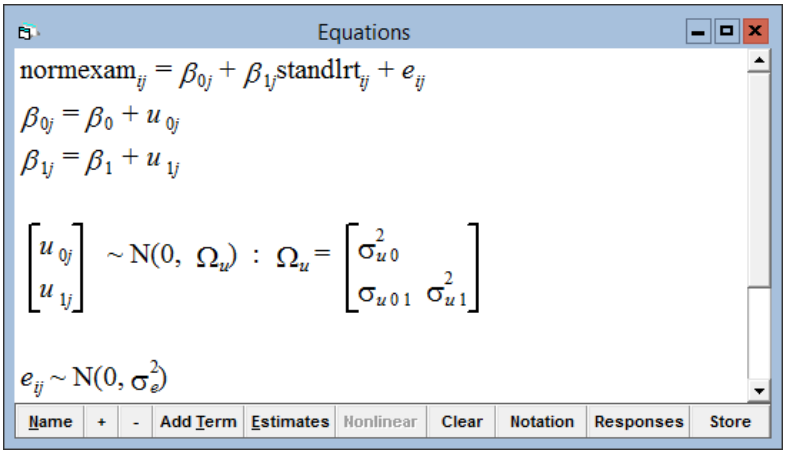
This model, which now includes 130 fixed effects, is clearly rather cumbersome. It amounts to fitting a separate linear regression to each school. We may believe, as is reasonable in this case, that our 65 schools are sampled from a larger population of schools. In the fixed effects model, all the schools are treated independently and there is nothing in the model to represent the fact that they are drawn from the same population. Typically, we wish to make inferences to the population of level 2 units (schools) from which we have drawn our sample of 65 schools. This type of inference is not available with the fixed effects model. Also, as we explained in the last chapter, the fixed effects model does not permit the addition of school level explanatory variables, which may be of crucial interest to us.

In the light of these shortcomings we will proceed with a multilevel model to investigate the question of whether the coefficient of **standlrt** varies across schools.

If we regard the schools as a random sample from a population of schools, then we wish to specify a coefficient of **standlrt** that is random at level 2. To do this we need to inform MLwiN that the coefficient of x_{1ij}, or $\textbf{standlrt}_{ij}$, should have the subscript j attached:

- Select **Equations** on the **Model** menu
- Click **Estimates** until β_0 etc. are displayed in black
- Click on β_0 and check the box labelled **j (school)** then click **Done**
- Click on β_1 and check the box labelled **j (school)** then click **Done**
- Remove **schgend** from the model by clicking on β_2 or β_3 in the **Equations** window
- Select **Delete term** from the window that appears

This produces the following result:


Equations
normexam$_{ij}$ = β_{0j} + β_{1j}standlrt$_{ij}$ + e_{ij}
β_{0j} = β_0 + u_{0j}
β_{1j} = β_1 + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u01} & \sigma^2_{u1} \end{bmatrix}$
e_{ij} ~ N(0, σ^2_e)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

Now both the intercept and the slope vary randomly across schools. Hence in the first line of the display both β_0 and β_1 have a j subscript. The second line states that the intercept for the jth school (β_{0j}) is given by β_0, the average intercept across all the schools, plus a random departure u_{0j}. Likewise, the third line states that the slope for the jth school (β_{1j}) is given by β_1, the average slope across all the schools, plus a random departure u_{1j}. The parameters β_0 and β_1 are the fixed part (regression) intercept and slope coefficients. They combine to give the average line across all students in all schools. The terms u_{0j} and u_{1j} are random departures from β_0 and β_1, or 'residuals' at the school level; they allow the jth school's summary line to differ from the average line in both its slope and its intercept.

A new term, Ω_u, now appears in the **Equations** window. The terms u_{0j} and u_{1j} follow a multivariate (in this case bivariate) Normal distribution with mean vector $\mathbf{0}$ and covariance matrix Ω_u. In this model, since we have two random variables at level 2, Ω_u is a 2 by 2 covariance matrix. The elements of Ω_u are:

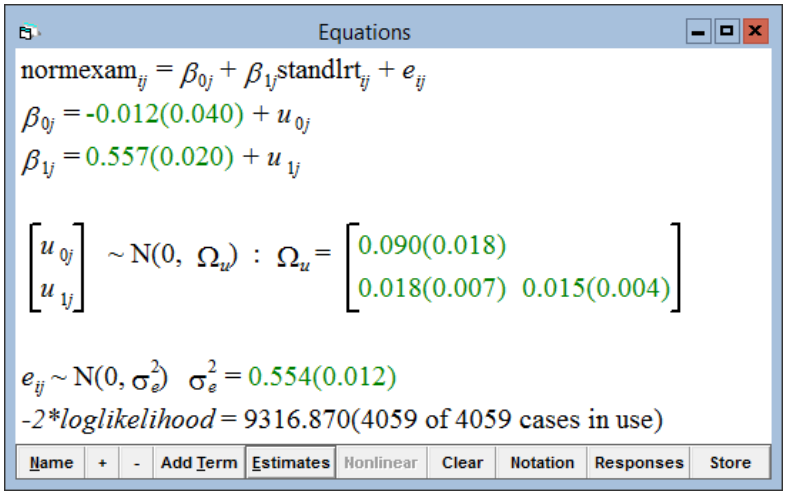
$\text{var}(u_{0j}) = \sigma^2_{u0}$	variation in the intercepts across the schools' summary lines
$\text{var}(u_{1j}) = \sigma^2_{u1}$	variation in the slopes across the schools' summary lines
$\text{cov}(u_{0j},u_{1j}) = \sigma_{u01}$	covariance between the school intercepts and slopes

Students' scores depart from their school's summary line by an amount e_{ij}, which as before are assumed to be normally distributed with mean 0 and variance σ^2_e. The letter u is used for random departures at level 2 (in this case school). The letter e is used for random departures at level 1 (in this case student).

To obtain estimates for this model, do the following:

- Click **More**
- Click **Estimates** (twice if necessary)

The result is as follows:


Equations
normexam$_{ij}$ = β_{0j} + β_{1j}standlrt$_{ij}$ + e_{ij}
β_{0j} = -0.012(0.040) + u_{0j}
β_{1j} = 0.557(0.020) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} 0.090(0.018) & \\ 0.018(0.007) & 0.015(0.004) \end{bmatrix}$
e_{ij} ~ N(0, σ_e^2) σ_e^2 = 0.554(0.012)
-2*loglikelihood = 9316.870(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

From line 3, we see that the estimate of β_1, the coefficient of **standlrt**, is 0.557 (standard error 0.020), which is close to the estimate obtained from the model with a single slope. However, the individual school slopes vary about this mean with a variance estimated as 0.015 (standard error 0.004). The intercepts of the individual school lines also differ. Their mean is -0.012 (standard error 0.040) and their variance is 0.090 (standard error 0.018). In addition, there is a positive covariance between intercepts and slopes estimated as $+0.018$ (standard error 0.007), suggesting that schools with higher intercepts tend to have steeper slopes; this corresponds to a correlation between the intercept and slope (across schools) of $0.018/\sqrt{0.015 \times 0.090} = 0.49$. This positive correlation will lead to a fanning out pattern when we plot the schools' predicted lines.

As in the previous model, the pupils' individual scores vary around their schools' lines by quantities e_{ij}, the level 1 residuals, whose variance is estimated as 0.554 (standard error 0.012).

Comparing this model with a single slope model (without school gender effects) you will see that -2 log-likelihood value has decreased from 9357.2 to 9316.9, a difference of 40.3. The new model involves two extra parameters, the variance of the slope residuals u_{1j} and their covariance with the intercept residuals u_{0j}. Therefore, the change in the -2 log-likelihood value (which is also the change in deviance) has a chi-squared distribution on 2 degrees of freedom under the null hypothesis that the extra parameters have population values of zero. The change is highly significant, confirming the better fit of the more elaborate model to the data.

4.5 Graphing predicted school lines from a random slope model

We can look at the pattern of the schools' summary lines by updating the predictions in the graph display window. We need to form the prediction equation

$$\hat{y} = \hat{\beta}_{0j}x_0 + \hat{\beta}_{1j}x_{1ij}$$

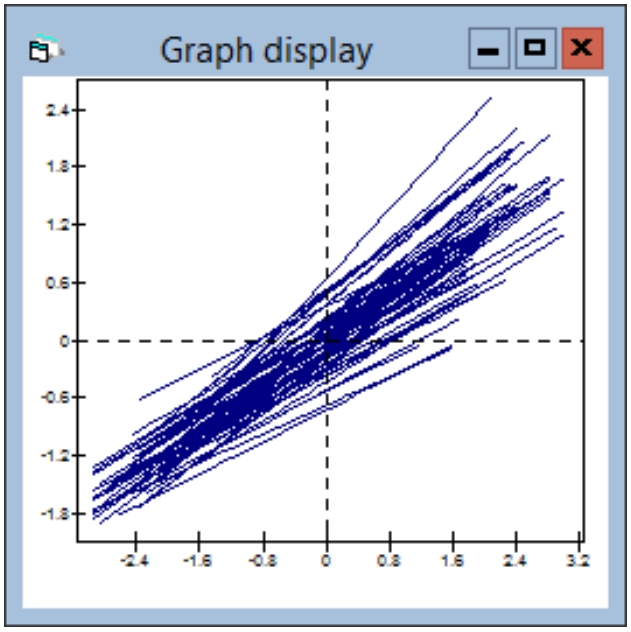
One way to do this is to:

- Select the **Model** menu
- Select **Predictions**
- In the **Predictions** window click on the word **variable**
- From the menu that appears choose **Include all explanatory variables**
- In the **output from prediction to** drop-down list, select **c11**
- Click **Calc**

This will overwrite the previous predictions from the random intercepts model with the predictions from the random slopes model. The graph window will be automatically updated. If you do not have the graph window displayed, then:

- Select the **Graphs** menu
- Select **Customised Graph(s)**
- Click **Apply**

The graph display window should look like this:



The graph shows the fanning out pattern for the school prediction lines that is implied by the positive intercept/slope covariance at the school level.

To test your understanding, try building different prediction equations in the **predictions** window. Before you press the **Calc** button, try and work out how the graph in the **graph display** window will change.

That concludes the fourth chapter. It is a good idea to save your worksheet using the **Save worksheet As** option on the **File** menu.

Note that saving a worksheet preserves the current contents of the data columns (including new ones you may have created) and saves the current model. Settings for graphs are saved; to see the graph(s) you have created again you will need to go to the ***Customised graph*** *window, select the appropriate display number, and click* ***Apply****.*

Chapter learning outcomes

- ⋆ What a random intercept model is
- ⋆ What a random slope model is
- ⋆ The equations used to describe these models
- ⋆ How to construct, estimate and interpret these models using the **Equations** window in MLwiN
- ⋆ How to carry out simple tests of significance

- ⋆ How the effect of clustering can distort the standard errors of OLS regression coefficients
- ⋆ How to use the **predictions** window to calculate predictions from the model estimates

Chapter 5

Graphical Procedures for Exploring the Model

5.1 Displaying multiple graphs

In Chapter 3 we produced graphical displays of the school level residuals in our random intercept model, using choices on the **Plots** tab of the **Residuals** window to specify the type of plot we wanted. MLwiN has very powerful graphical facilities, and in this chapter we shall see how to obtain more sophisticated graphs using the **Customised graph** window. We will also use some of these graphical features to explore the random intercepts and random slopes models.

Graphical output in MLwiN can be described (very appropriately) as having three levels. At the highest level, a *display* is essentially what can be displayed on the computer screen at one time. You can specify up to 10 different displays and switch between them, as you require. A display can consist of several *graphs*. A graph is a frame with x and y axes showing lines, points or bars, and each display can show an array of up to 5×5 graphs. A single graph can plot one or more *data sets*, each one consisting of a set of x and y coordinates held in worksheet columns.

To see how this works, we will calculate level 2 residuals for the random intercepts model we fitted in Section 4.1 and produce a caterpillar plot as a starting point for some graphical exploration of the model. Follow the instructions in Section 4.1 to set up and run the model again, then:

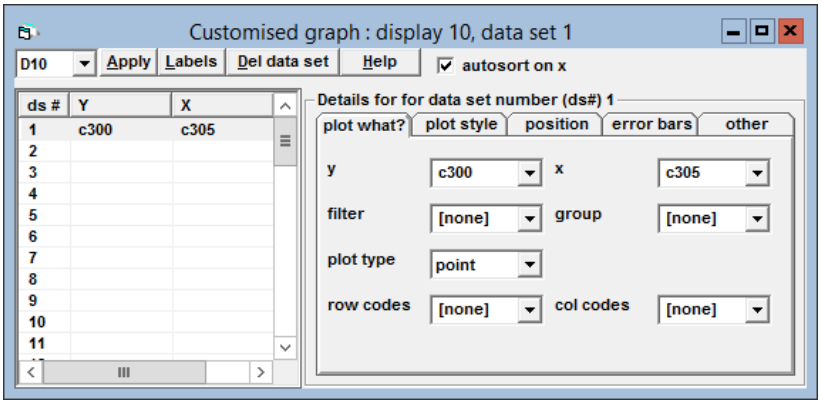
- Select **Residuals** on the **Model** menu
- Select the **Settings** tab of the **Residuals** window
- From the **level:** drop-down list, select **2:school**

- Click the **Set Columns** button
- Edit the **SD Multiplier** to be **1.96**
- Click **Calc**
- Click on the **Plots** tab
- Select **residual +/− 1.96 SD x rank**
- Click **Apply**

Notice that at the bottom right of the **Plots** tab of the **Residuals** window it says "Output to graph display number" and in the box beneath **D10** has been selected. This means that the specification for the caterpillar plot has been stored in Display 10. We can look at this specification and change or add things:

- Select **Customised Graph(s)** on the **Graphs** menu
- From the drop-down list at the top-left of the display select **D10**

The following window appears:



The display so far contains a single graph and this in turn contains a single data set, **ds1**, for which the y and x coordinates are in columns **c300** and **c305** respectively. As you can check from the **Residuals** window, these contain the level 2 residuals and their ranks.

Let us add a second graph to this display containing a scatter plot of **normexam** against **standlrt** for the whole of the data. First we need to specify this as a second data set.

- Select data set number 2 (**ds#2**) by clicking on the row labelled **2** in the grid on the left hand side of the window
- Use the y and x drop-down lists on the **plot what?** tab to specify **normexam** and **standlrt** as the y and x variables in **ds#2**.

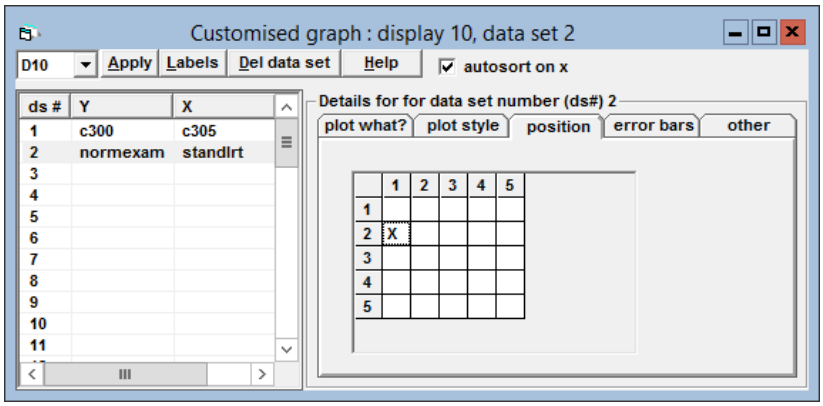
Next we need to specify that this graph is to be separate from the caterpillar plot. To do this:

- Click the **position** tab on the right hand side of the **Customised graph** window

The display can contain a 5×5 grid or trellis of different graphs. The cross in the position grid indicates where the current data set, in this case (**normexam**, **standlrt**), will be plotted. The default position is row 1, column 1. We want the scatter plot to appear vertically below the caterpillar plot in row 2, column 1 of the trellis, so

- Click the row 2 column 1 cell in the position grid

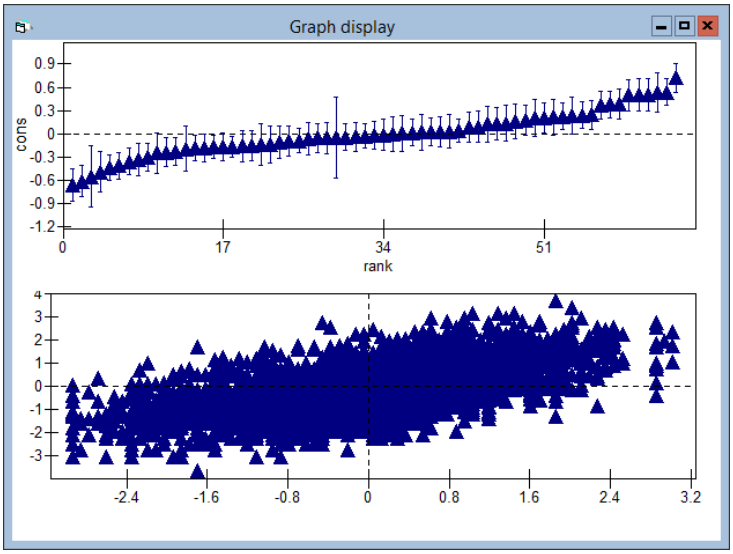
This looks as follows:



Now to see what we have got:

- Click the **Apply** button at the top of the **Customised graph** window

The following display will appear on the screen:



5.2 Highlighting in graphs

To illustrate the highlighting facilities of MLwiN let us add a third graph to our set — a replica of a graph we produced in Section 4.2 showing the 65 individual regression lines of the different schools. We will create and highlight the average line from which they depart in a random manner. We can insert this graph between the two graphs that we already have.

We need to calculate the points for plotting in the new graph. For the individual lines:

- Select **Predictions** on the **Model** menu
- Click on **Variable**
- Select **Include all explanatory variables**
- Click on e_{0ij} to remove it (it will turn from black to grey)
- In the **output from prediction to** list, select **c11**
- Click **Calc**

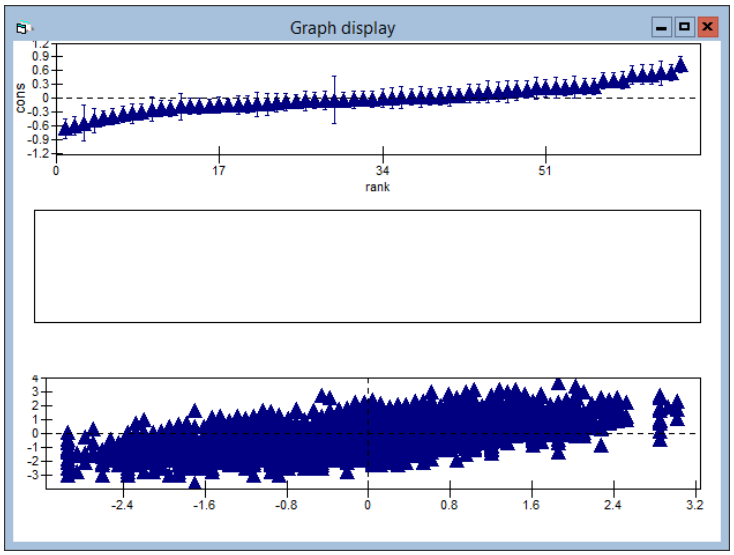
This will form the predictions using the level 2 (school) residuals but not the level 1 (student) residuals. For the overall average line we need to eliminate the level 2 residuals, leaving only the fixed part of the model:

- In the **Predictions** window, click on u_{0j} to remove it
- In the **output from prediction to** list, select **c12**
- Click **Calc**

The **Customised graph** window is currently showing the details of data set **ds#2**, the scatter plot. With this data set selected:

- Click on the **position** tab
- In the grid, click the cell in row 3, column 1
- Click **Apply**

The display now appears as follows:

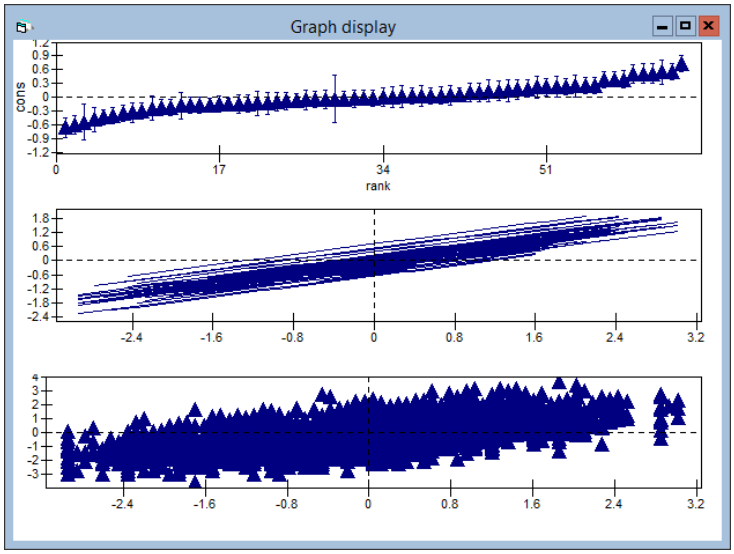


We have not yet specified any data sets for the middle graph, so it is blank for the time being. Here and elsewhere you may need to resize and re-position the graph display window by pulling on its borders in the usual way.

Now let us plot the lines that we have calculated. We need to plot **c11** and **c12** against **standlrt**. For the individual school lines we shall need to specify the group, meaning that the 65 lines should be plotted separately. In the **Customised graph** window:

- Select data set **ds#3** at the left of the window
- In the **y** drop-down list, specify **c11**, and in the **x** drop-down list, specify **standlrt**
- In the **group** drop-down list, select **school**
- In the **plot type** drop-down list, select **line**
- Select the **position** tab, and in the grid, click the cell in row 2, column 1
- Click **Apply**

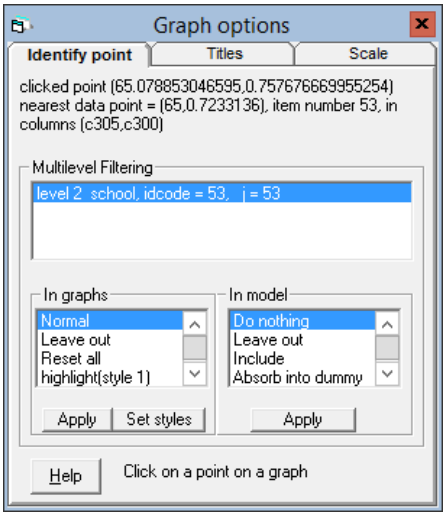
This produces the following display:



Now we can superimpose the overall average line by specifying a second data set for the middle graph. So that it will show up, we can plot it in black and make it thicker than the other lines:

- Select data set **ds#4** at the left of the **Customised graphs** window
- In the **y** drop-down list, specify **c12**, and in the **x** drop-down list, specify **standlrt**
- In the **plot type** drop-down list, select **line**
- Select the **plot style** tab
- In the **colour** drop-down list, select **0 black**
- In the **line thickness** drop-down list, select **3**
- Select the **position** tab, and in the grid, click the cell in row 2, column 1
- Click **Apply**

MLwiN makes it possible to zero in on features of particular schools that appear to be unusual in some way. To investigate some of this with the graphs that we have produced, click in the top graph on the point corresponding to the largest of the level 2 residuals, the one with rank 65. This brings up the following **Graph options** window:



The box in the centre shows that we have selected the 53rd school out of the 65, whose identifier is 53. We can *highlight* all the points in the display that belong to this school:

- Select **highlight(style 1)**
- Click **Apply**

You will see that the appropriate point in the top graph, two lines in the middle graph and a set of points in the scatter plot have all become coloured red. The individual school line is the thinner of the two highlighted lines in the middle graph. As would be expected from the fact that it has the highest intercept residual, the school's line is at the top of the collection of school lines.

It is not necessary to highlight all references to school 53. To de-highlight the school's contribution to the overall average line that is contained in data set **ds#4**:

- In the **Customised graph** window, select **ds#4**
- Click on the **other** tab
- Click on the **exclude from highlight** box
- Click **Apply**

In the caterpillar plot there is a residual around rank 30 that has very wide error bars. Let us try to see why. If you click on the point representing this school in the caterpillar plot, the **Graph options** window will identify it as school 48. Highlight the points belonging to this school in a different colour:

- Using the **In graphs** box of the **Graph options** window, select **highlight (style 2)**
- Click **Apply**

The points in the scatter plot belonging to this school will be highlighted in cyan, and inspection of the plot shows that there are only two of them. This means that there is very little information regarding this school. As a result, the confidence limits for its residual are very wide, and the residual itself will have been shrunk towards zero by an appreciable amount.

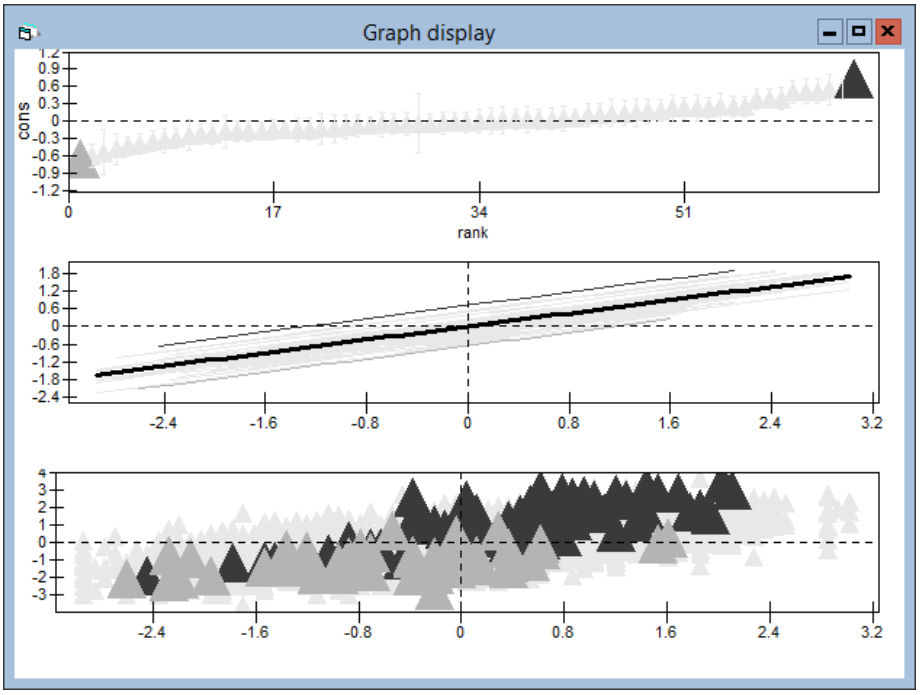
Next let us remove all the highlights from school 48.

- In the **In graphs** box of the **Graph options** window, select **Normal**
- Click **Apply**

Now let us look at the school at the other end of the caterpillar plot — the one with the smallest school level residual (it turns out to be school 59).

- Click on its point in the caterpillar plot
- In the **In graphs** box of the **Graph options** window, select **highlight (style 3)**
- Click **Apply**

The highlighting will remain and the graphical display will look something like this (we have changed some settings to aid visibility):



The caterpillar plot tells us simply that school 59 and 53 have different intercepts. One is significantly below the average line, and the other significantly above it. But the bottom graph suggests a more complicated situation. At higher levels of **standlrt**, the points for school 53 certainly appear to be consistently above those for school 59. But at the other end of the scale, at the left of the graph, there does not seem to be much difference between the schools. The graph indeed suggests that the two schools have different slopes, with school 53 having the steeper.

To follow up this suggestion, let us keep the graphical display while we extend our model to contain random slopes. To do this:

- From the **Model** menu select **Equations**
- Click on **Estimates** to show $\beta_{\mathbf{1}}$
- Click on $\beta_{\mathbf{1}}$ and check the box labelled **j(school)** to make it random at level 2
- Click **Done**
- Click **More** on the main toolbar and watch for convergence
- Close the **Equations** window

The results should match those of the last figure in Section 4.4.

Now we need to update the predictions in column **c11** to take account of the new model:

- From the **Model** menu select **Predictions**
- Click on $\mathbf{u_{0j}}$ and $\mathbf{u_{1j}}$ to include them in the predictions
- In the **output from prediction to** drop-down list, select **c11**
- Click **Calc**

Notice that the graphical display is automatically updated with the new contents of c11.

The caterpillar plot at the top of the display is now out of date, however, having been calculated from the previous model. (Recall that we used the Residuals window to create the caterpillar plot). We now have two sets of level 2 residuals, one giving the intercepts for the different schools and the other the slopes. To calculate and store these:

- Select **Residuals** from the **Model** menu
- Select **2:school** from the **level** drop-down list

- Edit the **start output at** box to **310**
- Edit the **SD Multiplier** to be **1.96**
- Click the **Set columns** button
- Click **Calc**

The intercept and slope residuals will be put into columns c310 and c311. To plot them against each other:

- In the **Customised graph** window select data set **ds#1** and click **Del data set**
- From the **y** drop-down list select **c310**
- From the **x** drop-down list select **c311**
- Click **Apply**

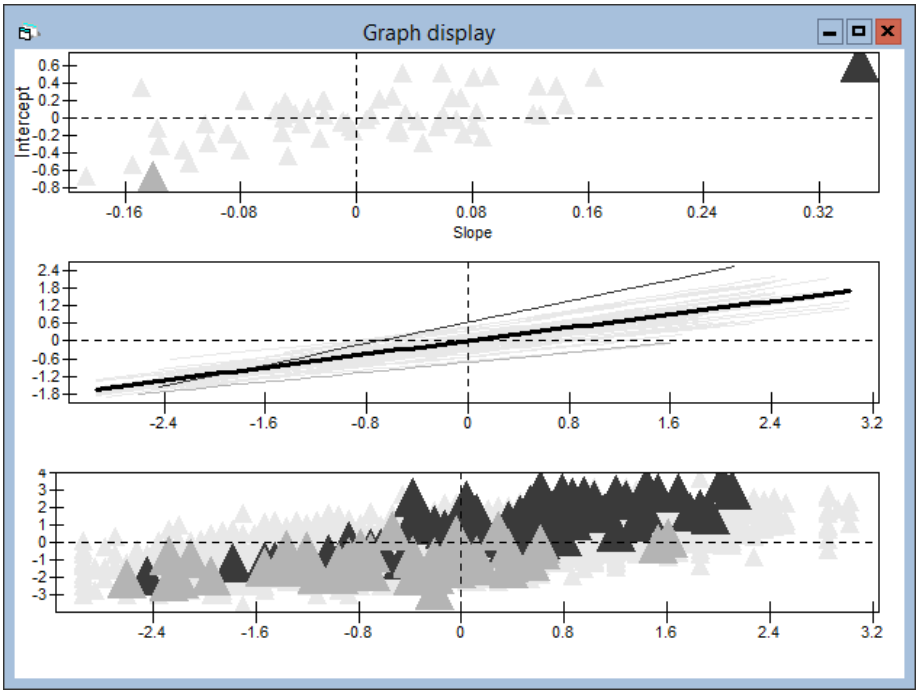
The axis titles in the top graph also need changing.

*Note that if you use the **Customised graph** window to create graphs then titles are not automatically added to the graphs. This is because a graph may contain many data sets so in general there is no obvious text for the titles. The existing titles appear because the original graph was constructed using the **Plots** tab on the **Residuals** window. You can add or alter titles by clicking on a graph.*

In our case:

- Click somewhere in the top graph to bring up the **Graph options** window
- Select the **Titles** tab
- Edit the **y title** to be **Intercept** and edit the **x title** to be **Slope**
- Click **Apply**

You can add titles to the other graphs in the same way if you wish. Now the graphical display will look like this:



The two schools at the opposite ends of the scale are still highlighted, and the middle graph confirms that there is very little difference between them when values of **standlrt** are small. School 53 stands out as exceptional in the top graph, with a high intercept and much higher slope than the other schools.

For a more detailed comparison between schools 53 and 49, we can put 95% confidence bands around their regression lines. To calculate the widths of the bands and plot them:

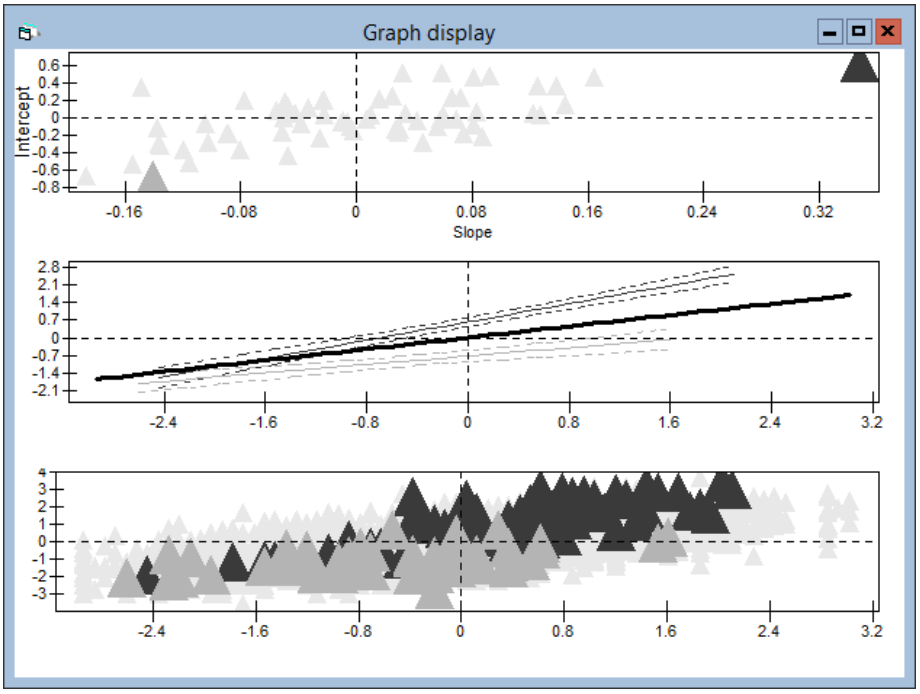
- Select **Predictions** from the **Model** menu
- In the box in the lower left corner of the **predictions** window, edit the multiplier of **S.E.** to 1.96
- From the **S.E. of** drop-down list select **level 2 resid. function**
- From the **output to** drop-down list select column **c13**
- Click **Calc**
- In the **Customised graph** window select data set **ds#3**
- Select the **error bars** tab
- From the **y errors +** list select **c13**
- From the **y errors −** list select **c13**
- From the **y error type** list select **lines**
- Click **Apply**

This draws 65 confidence bands around 65 school lines, which is not a particularly readable graph. However, we can focus in on the two highlighted

schools by drawing the rest in white.

- Select the **Customised graph** window
- Select data set **ds#3**
- From the **colour** list, on the **plot style** tab, select **15 white**
- Click **Apply**

The result is as follows:



The confidence bands confirm that what appeared to be the top and bottom schools cannot be reliably separated at the lower end of the intake scale.

Looking at the intercepts and slopes may shed light on interesting educational questions. For example, schools with large intercepts and small slopes — plotted in the top left quadrant of the top graph — are 'levelling up', i.e, they are doing well by their students at all levels of initial ability. Schools with large slopes are differentiating between levels of intake ability. The highlighting and other graphical features of MLwiN can be useful for exploring such features of complex data. See Yang et al. (1999) for a further discussion of this issue.

Chapter learning outcomes

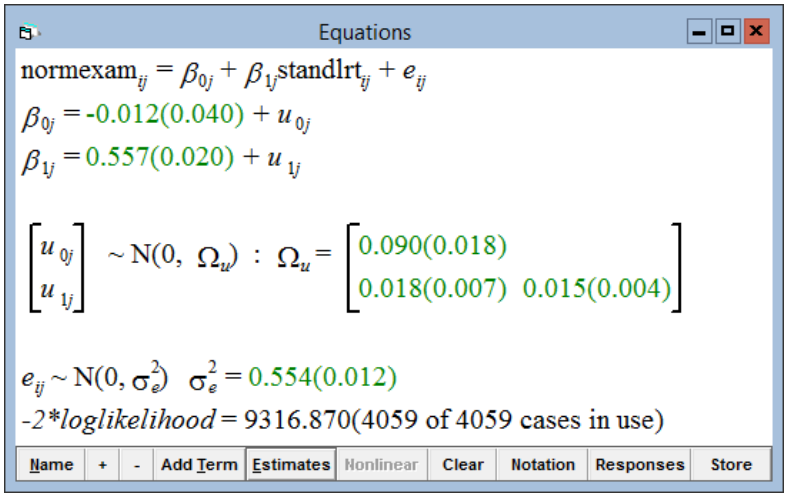
- ⋆ How to make different types of graphical presentations of complex data
- ⋆ How to explore features of multilevel data using graphical facilities such as highlighting
- ⋆ How to describe differences among higher level units (e.g. schools) when a random slopes model has been fitted, and in particular the fact that such differences cannot be expressed with a sinle number

Chapter 6

Contextual Effects

Many interesting questions in the social sciences are of the form "How are individuals affected by their social contexts?" For example, do girls learn more effectively in a girls' school or in a mixed sex school? Do low ability pupils fare better when they are educated alongside higher-ability pupils, or do they fare worse?

In this chapter we will develop models to investigate these two questions. Our starting point will be the model we fitted in Section 4.4.



To set up this model, do the following:

- Select the **Open Worksheet** option on the **File** menu
- Open the file **tutorial.ws**
- Select the **Equations** menu item from the **Model** menu
- Click the **Notation** button
- In the **Notation** window, clear the box beside **general**

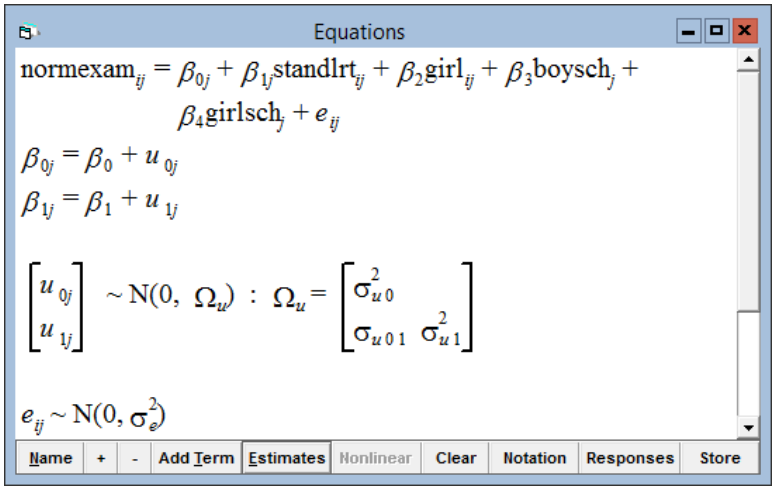
- Click **Done**
- Click on **y** in the **Equations** window
- In the **Y variable** window, select **y**: **normexam**
- Select **N levels: 2-ij**
- Select **level 2(j): school**
- Select **level 1(i): student**
- Click **Done**
- In the **Equations** window, click on β_0
- In the **intercept** window, check the **j(school)** checkbox
- Click **Done**
- Click on the **Add term** button
- From the **Specify term** window's **variable** drop-down list, select **standlrt**
- Click **Done**
- Click on β_1
- In the **X variable** window, check the **j(school)** checkbox
- Click **Done**
- Click the **Estimates** button twice
- Click **Start**

6.1 The impact of school gender on girls' achievement

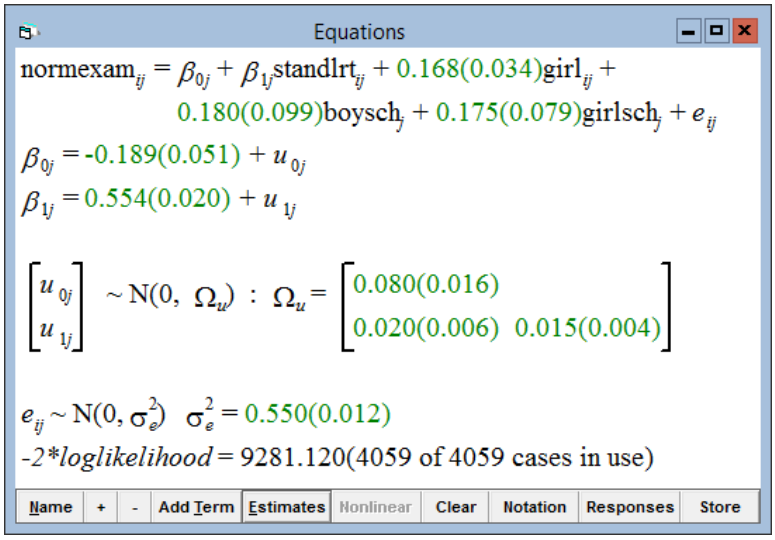
Let's add pupil gender and school gender effects into the above model:

- Click the **Add term** button
- From the **variable** drop-down list select **girl**
- Click **Done**
- Click the **Add term** button
- From the **variable** drop-down list select **schgend**
- Click **Done**

The **Equations** window should now look like this (after clicking the **Estimates** button):


Equations
normexam$_{ij}$ = β_{0j} + β_{1j}standlrt$_{ij}$ + $\beta_2$girl$_{ij}$ + $\beta_3$boysch$_j$ + $\beta_4$girlsch$_j$ + e_{ij}
β_{0j} = β_0 + u_{0j}
β_{1j} = β_1 + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u01} & \sigma^2_{u1} \end{bmatrix}$
e_{ij} ~ N(0, σ^2_e)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The reference category corresponds to boys in a mixed school. The dummy variable **girl** has subscript ij because it is a pupil level variable, whereas the two school level variables (**boysch** and **girlsch**) have only subscript j. We can run the model and view the results (by clicking on the **Estimates** button then on the **More** button on the main toolbar):


Equations
normexam$_{ij}$ = β_{0j} + β_{1j}standlrt$_{ij}$ + 0.168(0.034)girl$_{ij}$ + 0.180(0.099)boysch$_j$ + 0.175(0.079)girlsch$_j$ + e_{ij}
β_{0j} = -0.189(0.051) + u_{0j}
β_{1j} = 0.554(0.020) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} 0.080(0.016) & \\ 0.020(0.006) & 0.015(0.004) \end{bmatrix}$
e_{ij} ~ N(0, σ^2_e) σ^2_e = 0.550(0.012)
-2*loglikelihood = 9281.120(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The reference subgroup is "boys in a mixed school". We have four possible pupil subgroups. These are listed below, along with the corresponding explanatory variable pattern and model prediction for that group.

Pupil Subgroup	Values of Dummy Variables			Predicted Mean Value
	girl	**boy** sch	**girl** sch	
Boys in a mixed school	0	0	0	-0.189
Girls in a mixed school	1	0	0	-0.189 + 0.168
Boys in a boys' school	0	1	0	-0.189 + 0.180
Girls in a girls' school	1	0	1	-0.189 + 0.168 + 0.175

Girls in a mixed school do 0.168 of a standard deviation[1] better than boys in a mixed school. Girls in a girls' school do 0.175 points better than girls in a mixed school and (0.175 + 0.168) points better than boys in a mixed school. Boys in a boys' school do 0.18 points better than boys in a mixed school.

Adding these three parameters produced a reduction in the deviance of 35, which, under the null hypothesis of no effects, follows a chi-squared distribution with three degrees of freedom. You can look this probability up using the **Tail Areas** option on the **Basic Statistics** menu. The value is highly significant.

In the 2×3 table of gender by school gender there are two empty cells because there are no boys in a girls' school and no girls in a boys' school. We are therefore currently using a reference group and three parameters to model a four-entry table. Because of the empty cells the model is saturated, and no higher-order interactions can be added.

The pupil gender and school gender effects modify the intercept (interpreted when **standlrt** = 0). An interesting question is whether these effects change across the intake score spectrum. To address this we need to extend the model to include the interaction of the continuous variable **standlrt** with our categorical pupil and school level gender variables. Let's do this for the school gender variable first.

- Click the **Add Term** button on the **Equations** window
- In the **order** box of the **Specify term** window, select **1** (for a first order interaction)
- In the upper **variable** list box, select **schgend**
- In the lower **variable** list box, select **standlrt**
- Click **Done**

The **Equations** window will be automatically modified to include the two new interaction terms. Run the model by pressing **More** on the main toolbar.

The deviance reduces by less than one unit. From this we conclude there is no evidence of an interaction between the school gender variables and intake score. You can verify that the same is true for the interaction of pupil gender and intake score. Remove the school gender by intake score interaction as follows:

- Click on either of the interaction terms (**boysch.standlrt** or **girlsch.standlrt**)

[1] Recall that the **normexam** variable has been normalized to have a mean of 0 and a standard deviation of 1 in the full sample, so predicted effects of pupil gender and school gender will be in standard deviation units.

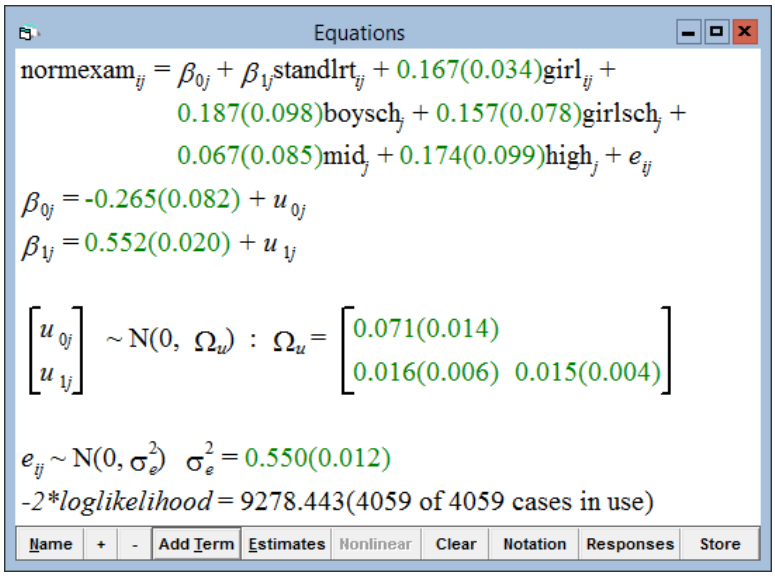
- In the **X variable** window, click on the **delete Term** button
- You will be asked if you want to remove the two terms in the **schgend * standlrt** interaction. Click **Yes**

6.2 Contextual effects of school intake ability averages

The variable **schav** was constructed by first computing the average intake ability (**standlrt**) for each school. Then, based on these averages, the bottom 25% of schools were coded 1 (**low**), the middle 50% were coded 2 (**mid**) and the top 25% were coded 3 (**high**). Let's include the two dummy variables for this categorical school level contextual variable in the model.

- Click the **Add Term** button in the **Equations** window
- In the **variable** list box on the **Specify term** window, select **schav**
- Click **Done**

Run the model by clicking the **More** button. The **Equations** window will now look like this:


Equations
normexam$_{ij}$ = β_{0j} + β_{1j}standlrt$_{ij}$ + 0.167(0.034)girl$_{ij}$ +
0.187(0.098)boysch$_j$ + 0.157(0.078)girlsch$_j$ +
0.067(0.085)mid$_j$ + 0.174(0.099)high$_j$ + e_{ij}
β_{0j} = -0.265(0.082) + u_{0j}
β_{1j} = 0.552(0.020) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} 0.071(0.014) & \\ 0.016(0.006) & 0.015(0.004) \end{bmatrix}$
e_{ij} ~ N(0, σ_e^2) σ_e^2 = 0.550(0.012)
-2*loglikelihood = 9278.443(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

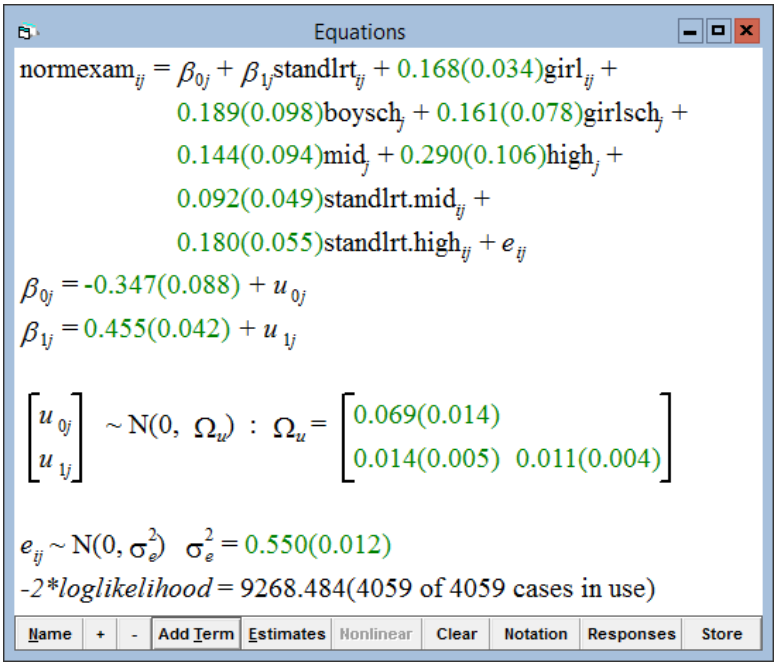
Pupils in the low ability schools are the reference group. Children attending **mid** and **high** ability schools score 0.067 and 0.174 points, respectively, more than reference group children. These effects are of borderline statistical significance, however.

Note that the deviance has been reduced by just 2.7 (9281.12 − 9278.44) *compared with the model involving* ***standlrt****, pupil gender and school gender. This change, when compared to a chi squared distribution with two degrees of freedom is not significant.*

This model assumes the contextual effects of school ability are the same across the intake ability spectrum because these contextual effects are modifying just the intercept term. That is the effect of being in a **high** ability school is the same for low ability and high ability pupils. To relax this assumption we need to include the interaction between **standlrt** and the school ability contextual variables. To do this:

- Click on the **Add Term** button
- In the **order** box of the **Specify term** window select **1**
- Select **standlrt** in the top **variable** list box
- Select **schav** in the lower **variable** list box
- Click **Done**
- Run the model by clicking the **More** button

The model converges to:


Equations
normexam$_{ij}$ = β_{0j} + β_{1j}standlrt$_{ij}$ + 0.168(0.034)girl$_{ij}$ +
0.189(0.098)boysch$_j$ + 0.161(0.078)girlsch$_j$ +
0.144(0.094)mid$_j$ + 0.290(0.106)high$_j$ +
0.092(0.049)standlrt.mid$_{ij}$ +
0.180(0.055)standlrt.high$_{ij}$ + e_{ij}
β_{0j} = -0.347(0.088) + u_{0j}
β_{1j} = 0.455(0.042) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} 0.069(0.014) & \\ 0.014(0.005) & 0.011(0.004) \end{bmatrix}$
e_{ij} ~ N(0, σ_e^2) σ_e^2 = 0.550(0.012)
-2*loglikelihood = 9268.484(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The slope coefficient for **standlrt** for pupils from **low** intake ability schools is 0.455. For pupils from **mid** ability schools the slope is steeper 0.455 + 0.092 and for pupils from **high** ability schools the slope is steeper still 0.455 + 0.18. These two interaction terms have explained variability in the slope of **standlrt** in terms of a school level variable, therefore the between-school

variability of the **standlrt** slope has been substantially reduced (from 0.015 to 0.011).

*Note that the previous contextual effects **boy** sch, **girl** sch, **mid** and **high** all modified the intercept and therefore fitting these school level variables reduced the between school variability of the intercept (σ^2_{u0}).*

We now have three different linear relationships between the output score (**normexam**) and the intake score (**standlrt**) for pupils from **low**, **mid** and **high** ability schools. The prediction line for boys in mixed **low** ability schools is

$$\hat{\beta}_0 \text{cons} + \hat{\beta}_1 \mathbf{standlrt}_{ij}$$

The prediction line for boys in mixed **high** ability schools is

$$\hat{\beta}_0 \text{cons} + \hat{\beta}_1 \mathbf{standlrt}_i j + \hat{\beta}_6 \text{high}_j + \hat{\beta}_8 \mathbf{standlrt}.\text{high}_{ij}$$

The difference between these two lines, that is the effect of being in a **high** ability school (regardless of pupil and school gender) is

$$\hat{\beta}_6 \text{high}_j + \hat{\beta}_8 \mathbf{standlrt}.\text{high}_{ij}$$

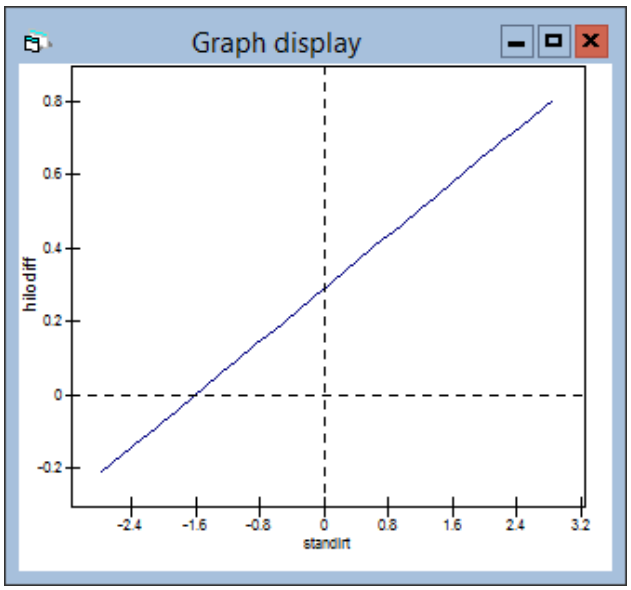
We can create this prediction function to examine the impact of school ability on students of different abilities:

- On the **Model** menu, select **Predictions**
- Click in turn on β_6, β_8
- In the output from prediction to list, select **c30**
- Press Ctrl-N and rename **c30** to **hilodiff**
- Click **Calc**

We can plot this function as follows:

- Select the **Customised graph** window
- In the box in the upper left corner, select a new display, say **D5**
- In the **y** list select **hilodiff**
- In the **x** list select **standlrt**
- In the **plot type** list select **line**
- In the **filter** list select **high**
- Click **Apply**
- Click anywhere in the graph that appears
- Select the **Titles** tab of the **Graph** options window
- Type **hilodiff** in the **y title** box and **standlrt** in the **x title** box
- Click **Apply**

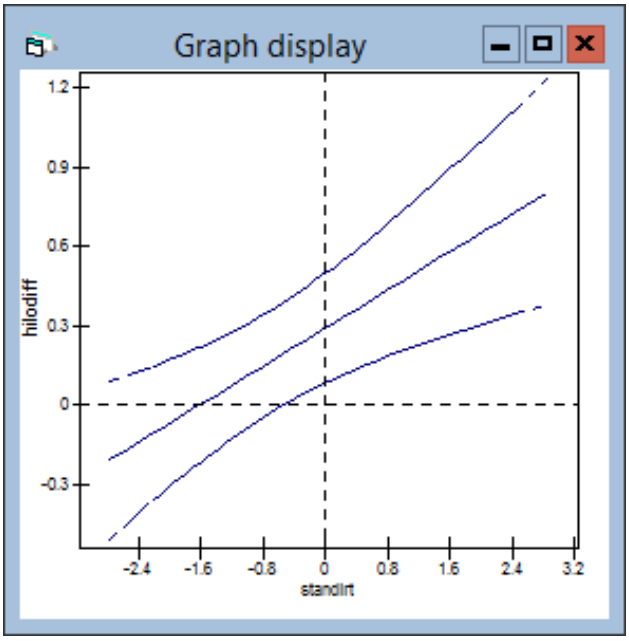
This graph (below) shows how the effect of being in a **high** ability school varies across the intake spectrum. On average, very able pupils being educated in a **high** ability school score 0.9 of a standard deviation higher in their outcome score than they would if they were educated in a **low** ability school. Pupils with intake scores below −1.7 fare better in **low** ability schools, i.e, **hilodiff** takes more negative values as **standlrt** drops further below this threshold. This finding has some educational interest but we will not pursue that here.



We can put a 95% confidence band around this line by doing the following:

- Select the **predictions** window
- Change the multiplier of **S.E. of** from 1.0 to **1.96**
- In the **S.E. of** list select **fixed**
- In the corresponding **output to** list select **c31**
- Click **Calc**
- Select the **Customised graph** window
- Select the **error bars** tab
- In the **y errors** + list select **c31**
- In the **y errors** − list select **c31**
- In the **y error type** list select **lines**
- Click **Apply**

This produces



Save your worksheet.

Chapter learning outcomes

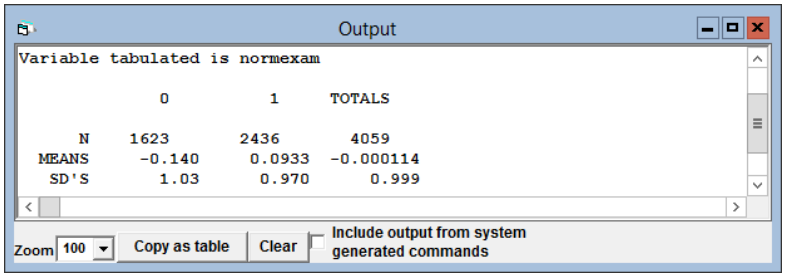
- ⋆ What is meant by contextual effects
- ⋆ How to set up multilevel models with interaction terms
- ⋆ How to include confidence regions around predictions

Chapter 7

Modelling the Variance as a Function of Explanatory Variables

7.1 A level 1 variance function for two groups

Back in Chapter 2 we tabulated **normexam** by gender and saw the following results:

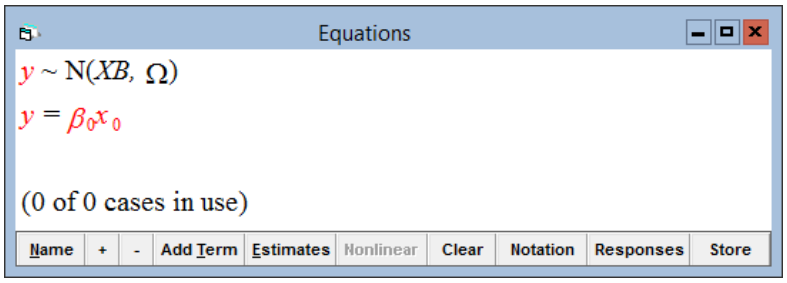


We observed that the SD of the **normexam** scores for girls (coded 1) is lower than the SD for boys. Until now all the models we have used have fitted a single random term at level 1 that assumes constant (homogenous) level 1 variation. We may want to fit a model that replicates this table, that is to directly estimate the means for boys and girls and to estimate separate student level variances for each group. The notation we have been using so far does not allow this because it assumes a common intercept β_0 and a single set of student level residuals e_i with a common variance σ^2_e. We need to use a more flexible notation to build this model.

- Open the file **tutorial.ws**

In this chapter we do not switch to simple notation mode. The **Equations**

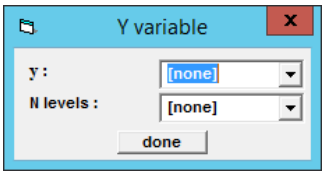
window with no model specified, with general notation mode looks like this:



A new first line is added stating that the response variable is Normally distributed. We now have the flexibility to specify alternative distributions for our response. We will explore these possibilities in later chapters.

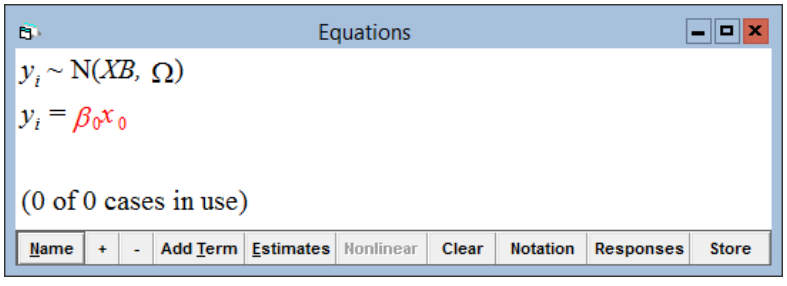
- Click on the red **y** in the **Equations** window

The following window will appear:



- From the **y** drop-down list select **normexam**
- From the **N levels** drop-down list, select **1-i**
- From the **level 1(i)** drop-down list that appears, select **student**
- Click **Name**

The **Equations** window should look like this:



Notice that with this more general notation, the β_0 coefficient has an explanatory variable x_0 associated with it. The value that x_0 takes determines the meaning of the β_0 coefficient. For example, if x_0 was a vector of 1s then β_0 would estimate an intercept common to all individuals. In the absence of other predictors, this would be an estimate of the overall mean. However, if x_0 contained a dummy variable, say 1 for boys and 0 for girls, then β_0 would

estimate the mean for boys. In the **Equations** window $\beta_0 x_0$ is coloured red, indicating we have not yet assigned a variable to x_0.

Recall that in our current model we do not want a common intercept; we want separate terms for the boy and girl means. We can achieve this by entering a dummy variable for boys and a second dummy variable for girls. First, let's create the boy dummy variable:

- On the **Data Manipulation** menu, select **Command interface**
- In the box at the bottom of the **Command interface** window, type the following commands:

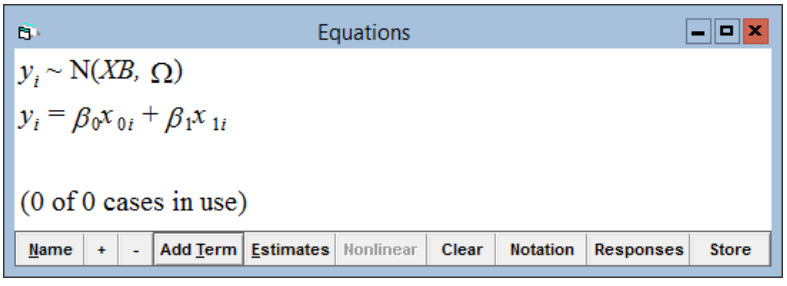
```
► calc c12 = 1 - 'girl'
► name c12 'boy'
```

- Note: click **Enter** after typing each command and include the quotation marks around the column names (boy and girl)

Now add the dummy variables to the model.

- Click the **Add Term** button in the **Equations** window
- In the **variable** list box on the **Specify term** window, select **boy**
- Click **Done**
- Use the same sequence of steps to enter the **girl** dummy into the model

The Equations window now looks like this:



We now have two explanatory variables:

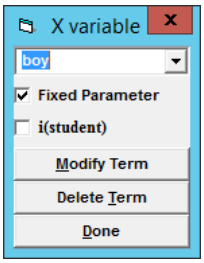
x_{0i}: which is 1 if the ith student is a boy
x_{1i}: which is 1 if the ith student is a girl

Coefficients β_0 and β_1 will estimate the means for boys and girls, respectively.

The next step is to introduce terms in the model for estimating separate variances for both groups. To do this

- Click on the term $\beta_0 x_{0i}$ in the **Equations** window

The following window appears:



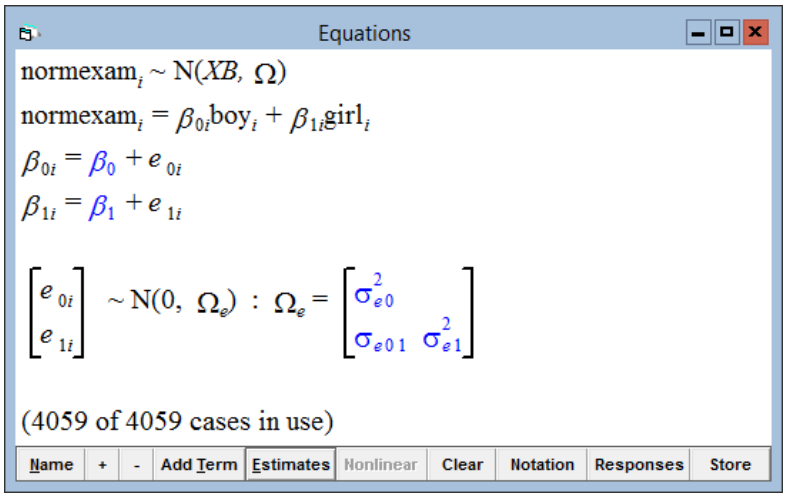
To estimate a student level variance for boys:

- Check the box labelled **i(student)**
- Click **Done**

Repeat this procedure for girls:

- Click on the term $\beta_1 x_{1i}$ in the **Equations** window
- In the **X variable** window, check the box labelled **i(student)**
- Click **Done**
- Click the **Name** button in the **Equations** window
- Click the **Estimates** button in the **Equations** window

The **Equations** window should now look like this:


Equations
normexam$_i$ ~ N(XB, Ω)
normexam$_i$ = β_{0i}boy$_i$ + β_{1i}girl$_i$
$\beta_{0i} = \beta_0 + e_{0i}$
$\beta_{1i} = \beta_1 + e_{1i}$
$\begin{bmatrix} e_{0i} \\ e_{1i} \end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} \sigma^2_{e0} & \\ \sigma_{e01} & \sigma^2_{e1} \end{bmatrix}$
(4059 of 4059 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

Both β_0 and β_1 now have i subscripts. Let's examine the second line in the **Equations** window:

$$\text{normexam}_i = \beta_{0i}\text{boy}_i + \beta_{1i}\text{girl}_i$$

a little more closely. If the ith response is for a boy, then the value of girl_i is zero and the second term on the right hand side disappears. Thus the boys' responses are modelled by the function $\beta_0 + e_{0i}$, where β_0 estimates the boys' mean. Conversely the girls' responses will be modelled by the function $\beta_1 + e_{1i}$, where β_1 estimates the girls' mean

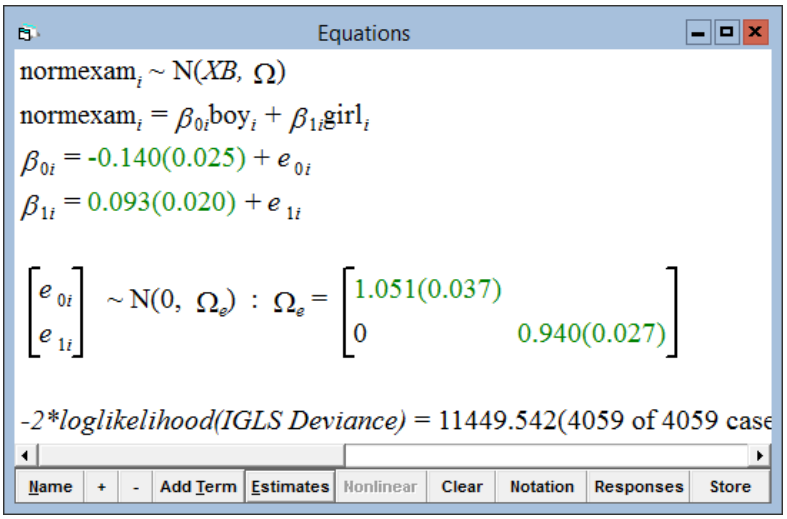
The departures of the boys' scores around their mean are given by the set of residuals e_{0i}. The departures of the girls' scores around their mean are given by a separate set of residuals e_{1i}. The variance of the boys' residuals is $\text{var}(e_{0i}) = \sigma^2_{e0}$ and the variance of the girls' residuals is $\text{var}(e_{1i}) = \sigma^2_{e1}$. These relationships can be found in the bottom lines of the display, which give the structure of the student level distributional assumptions. This part of display resembles what we saw when we had two residuals (intercept and slope) at the school level. The term σ_{e01} therefore specifies the covariance at the student level between the boy residuals and the girl residuals. However, this covariance can only exist if some students are both boys and girls. This is impossible, so we will remove the covariance term from the model. To do this:

- Click on the term σ_{e01} in the **Equations** window
- Click on the **Yes** button in the pop-up window that asks whether to remove the term

We are now ready to run the model.

- Click the **Start** button
- Click **Estimates**

The results are as follows:



The boys' and girls' means agree exactly with the tabular output at the top of the chapter. The table quotes SDs for the two groups; from the **Equations** window we have the SD for girls as $\sqrt{0.94} = 0.97$ and $\sqrt{1.05} = 1.03$ for boys.

This all may seem like a lot of work to replicate a simple table. However, the payoff is that when we work within the modelling framework offered in the **Equations** window, many extensions are possible that are well beyond the scope of simple tables.

We have modelled the student level variance as a function of gender. The function is

$$\text{var}(y_i) = \sigma^2_{e0} x_{0i} + \sigma^2_{e1} x_{1i} \tag{7.1}$$

where x_{0i} is 1 if the ith student is a boy and 0 if the ith student is a girl. Likewise, x_{1i} is 1 if the ith student is a girl and 0 if the ith student is a boy.

Equation (7.1) simplifies to σ^2_{e0} for boys, since for boys x_{0i} is always 1 and x_{1i} is always 0. Conversely, (7.1) simplifies to σ^2_{e1} for girls. It is instructive to look at how we arrive at the functional form in (7.1). Our current model is

$$\begin{aligned} y_i &= \beta_{0i} x_{0i} + \beta_{1i} x_{1i} \\ \beta_{0i} &= \beta_0 + e_{0i} \\ \beta_{1i} &= \beta_1 + e_{1i} \end{aligned}$$

which can be rewritten as

$$y_i = \beta_0 x_{0i} + \beta_1 x_{1i} + e_{0i} x_{0i} + e_{1i} x_{1i} \tag{7.2}$$

What is the student level variation? It is the variance of any terms in the model that contain student level residuals, that is the last two terms in equation (7.2). Using basic theory[1] about the variance of a linear combination of random variables, we can express the student level variation as:

$$\begin{aligned} \text{var}(y_i) &= \text{var}(e_{0i} x_{0i} + e_{1i} x_{1i}) \\ &= \text{var}(e_{0i} x_0) + 2\text{cov}(e_{0i} x_{0i}, e_{1i} x_{1i}) + \text{var}(e_{1i} x_{1i}) \\ &= \text{var}(e_{0i}) x^2_{0i} + 2\text{cov}(e_{0i}, e_{1i}) x_{0i} x_{1i} + \text{var}(e_{1i}) x^2_{1i} \\ &= \sigma^2_{e0} x^2_{0i} + 2\sigma_{e01} x_{0i} x_{1i} + \sigma^2_{e1} x^2_{1i} \end{aligned} \tag{7.3}$$

[1] See, for example, Kendall & Stewart (1997)

In our example σ_{e01} is set to zero, because a student cannot be both a boy and a girl, i.e, no student has both residuals. Also x_{0i} and x_{1i} are (0,1) variables therefore $x_{0i}^2 = x_{0i}$ and $x_{1i}^2 = x_{1i}$. The variance function in (7.3) therefore simplifies to the variance function (7.1).

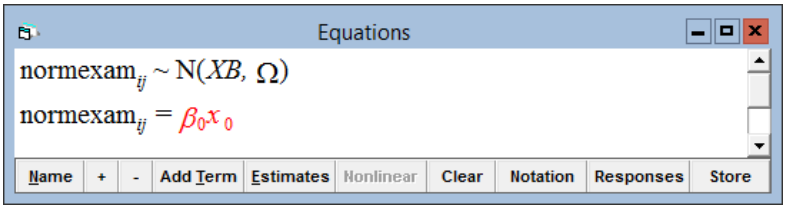
The notion of variance functions is a powerful one and is not restricted to level 1 variances. Let's look at the school level random intercept and slope model we fitted in Chapter 4 from the point of view of variance functions.

7.2 Variance functions at level 2

Let's set up the random slopes and intercepts model again. In the **Equations** window:

- Click the **Clear** button
- Click on **y**
- From the **y** drop-down list, select **normexam**
- From the **N-levels** drop-down list, select **2-ij**
- From the **level 2(j)** drop-down list, select **school**
- From the **level 1(i)** drop-down list, select **student**

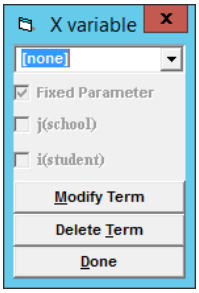
The **Equations** window now looks like this:



We are in the general notation mode therefore the β_0 coefficient has an explanatory variable x_0 associated with it. To specify a common intercept we will define x_0 as a constant vector of 1s. The column called **cons** in the worksheet contains such a vector of 1s, i.e, every pupil's value for **cons** is 1. To specify the random slopes and intercepts model, we begin by creating an intercept that is random at both levels:

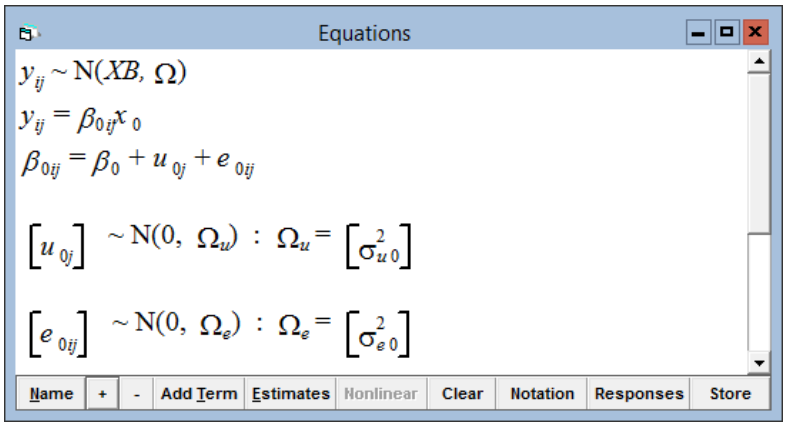
- Click on x_0 in the **Equations** window

The **X variable** window appears:



- From the drop-down list select **cons**
- Tick the **j(school)** check box
- Tick the **i(student)** check box

After you click on the **Name** button, followed by the + button twice, the **Equations** window now displays:



This is the first multilevel model we fitted back at the end of Chapter 2, written out in the more general notation. In Chapter 2 we wrote

$$\begin{aligned} y_{ij} &= \beta_{0j} + e_{ij} \\ \beta_{0j} &= \beta_0 + u_{0j} \end{aligned} \tag{7.4}$$

Substituting the second line of (7.4) into the first we have

$$y_{ij} = \beta_0 + u_{0j} + e_{ij} \tag{7.5}$$

Taking the second and third lines from the current **Equations** window we have

$$\begin{aligned} y_{ij} &= \beta_{0ij} x_0 \\ \beta_{0ij} &= \beta_0 + u_{0j} + e_{0ij} \end{aligned} \tag{7.6}$$

Substituting the second line of (7.6) into the first, we have

$$y_{ij} = \beta_0 x_0 + u_{0j} x_0 + e_{0ij} x_0 \tag{7.7}$$

Given x_0 is a vector of 1s we see that (7.7) is identical to (7.5).

> *Note that in* (7.7) *the student level residuals are given an additional 0 subscript. This indicates that these residuals are attached to explanatory variable* x_0. *This additional numbering, as we discussed earlier allows for further sets of student level residuals attached to other explanatory variables to be added to the model.*

We can now continue to add the slope term to the model.

- Click on the **Add Term**, opening the **Specify term** window
- Select **standlrt** from the **variable** drop-down list, and click **Done**

To allow the slope to vary randomly across schools:

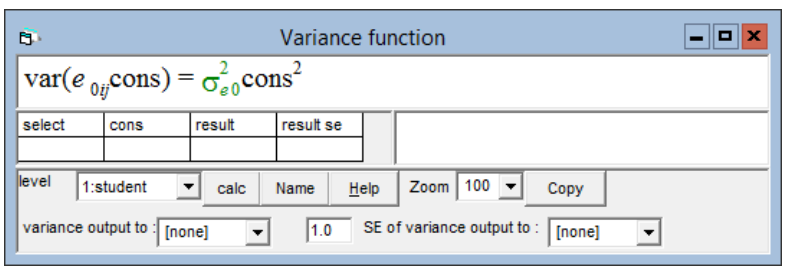
- Click on the term $\beta_1 x_1$ in the **Equations** window
- In the **X variable** window, tick the **j(school)** check box
- Click the **Done** button
- Click **Start**

We have now re-established the random slopes and intercept model. Remember that our aim is to explore level 2 variation from the variance function perspective. In Chapters 4 and 5 we saw a fanning out pattern of the school summary lines which tells us that schools are more variable for students with higher levels of **standlrt**. Another way of saying this is that the between-school variance is a function of **standlrt**.

Using the general notation in MLwiN we always specify the random variation in terms of coefficients of explanatory variables. The total variance at each level is thus a function of these explanatory variables. These functions are displayed in the **Variance function** window.

- On the **Model** menu, select **Variance function**
- Click **Name** button in the **Variance function** window

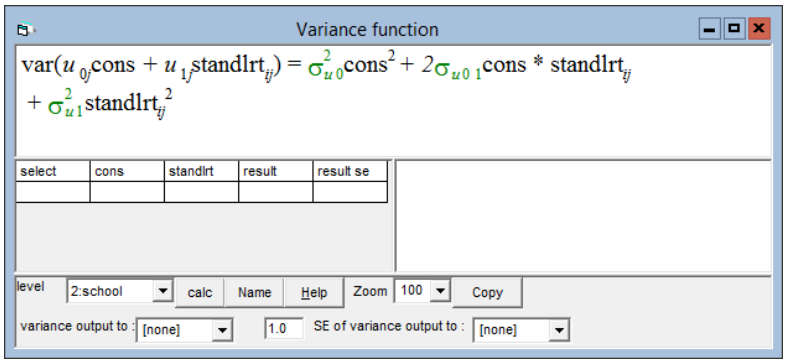
The initial display in this window is of the level 1 variance.



In the present model we have simple (constant) variation at level 1, as the above equation shows. Now look at the school level variation:

- In the **level** drop-down list, select **2:school**

We get the following:



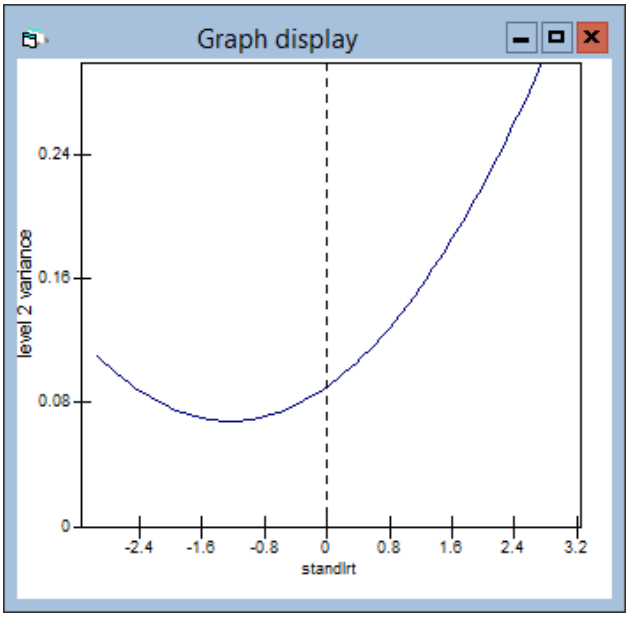
The function shown is simply the variance of the sum of two random coefficients times their respective explanatory variables, u_{0j}**cons** and u_{1j}**standlrt**$_{ij}$, written out explicitly. This has the same form as the student level variance function (7.3) that we derived earlier in the chapter, except es have now been replaced by us as we are operating at level 2 not level 1. Given that **cons** is a vector of ones, we see that the between-school variance is a quadratic function of **standlrt** with coefficients formed by the set of level 2 random parameters. The intercept in the quadratic function is σ^2_{u0}, the linear term is $2\sigma_{u01}$ and the quadratic term is σ^2_{u1}. We can compute this function and the **Variance function** window provides us with a simple means of doing this.

The columns in the window headed **select**, **cons**, **standlrt** and **result** are for computing individual values of the variance function. Since **standlrt** is a continuous variable it will be useful to calculate the level 2 variance for every value of **standlrt** that occurs.

- In the **variance output to** list on the tool bar, select **c30**
- Click **calc**

Now you can use the **Customised graph** window to plot **c30** against **standlrt**. The resulting graph (shown below) has had the y-axis rescaled to run between 0 and 0.3. To do this, click anywhere in the **Graph display** window, then click on the **Scale** tab of the **Graph Options** window. Check **User defined scale**, then change **ymin** to **0** and **ymax** to **0.3** and click **Apply**.

The apparent pattern of greater variation between schools for students with extreme **standlrt** scores, especially high ones, is consistent with the plot of prediction lines for the schools we viewed earlier.



We need to be careful about the interpretation of such plots. Polynomial functions are often unreliable at extremes of the data to which they are fitted. Another difficulty with using polynomials to model variances is that, for some values of the explanatory variables, they may predict a negative overall variance. To overcome this we can use nonlinear (negative exponential) functions to model variance. This is an advanced topic, and for details see Yang et al. (1999). However, we see that schools are more variable for students with high **standlrt** scores. This corresponds to the fanning out pattern of the school summary lines.

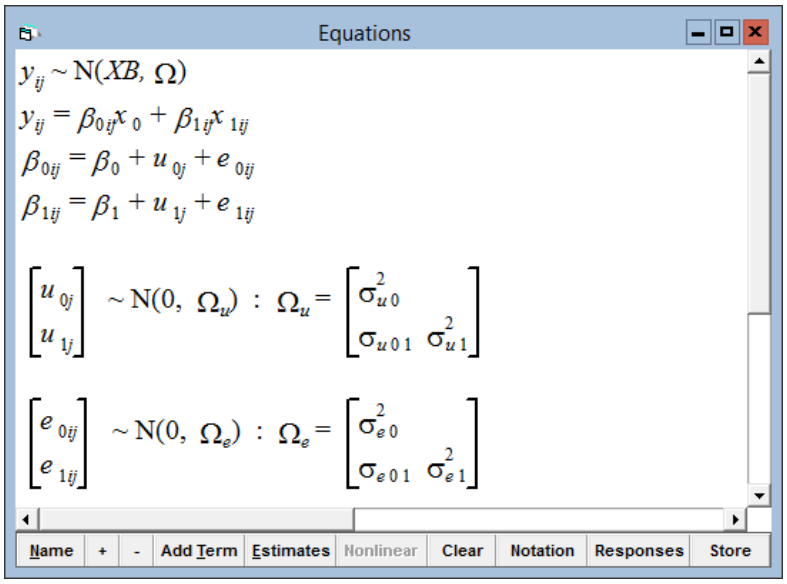
7.3 Further elaborating the model for the student-level variance

We have already seen how to model student level variation as a function of student gender. It might also be the case that the level 1 variation changes as a function of **standlrt**. That is the magnitude of the departures of students around their school's summary line changes in some systematic way with respect to **standlrt**.

Let's look and see if the student level variance changes as a function of **standlrt**. To do this we need to make the coefficient of **standlrt** random at the student level:

- In the **Equations** window click on β_1
- In the **X variable** window check the box labelled **i(student)** and click **Done**

This produces the following:

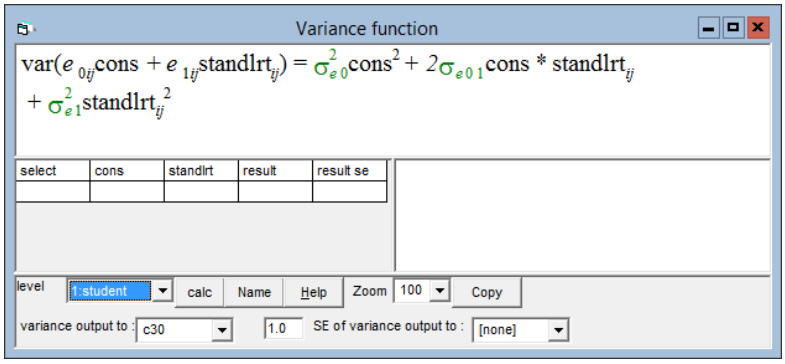

Equations
$y_{ij} \sim N(XB, \Omega)$
$y_{ij} = \beta_{0ij}x_0 + \beta_{1ij}x_{1ij}$
$\beta_{0ij} = \beta_0 + u_{0j} + e_{0ij}$
$\beta_{1ij} = \beta_1 + u_{1j} + e_{1ij}$
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u01} & \sigma^2_{u1} \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \\ e_{1ij} \end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} \sigma^2_{e0} & \\ \sigma_{e01} & \sigma^2_{e1} \end{bmatrix}$
Name
+
-
Add Term
Estimates
Nonlinear
Clear
Notation
Responses
Store

Now β_1 has a school level random term u_{1j} and a student level random term e_{1ij} attached to it. As we have seen, we can think of the variance of the u_{1j} terms, which is σ^2_{u1}, in two ways. Firstly, we can think of it as the between-school variation in the slopes. Secondly we can think of it as a coefficient in a quadratic function that describes how the between-school variation changes with respect to **standlrt**. Both conceptualisations are useful.

The situation at the student level is different. It does not make sense to think of the variance of the e_{1ij}s, that is σ^2_{e1}, as the between-student variation in the slopes. This is because a student corresponds to only one data point, and it is not possible to have a slope through one data point. However, the second conceptualisation where σ^2_{e1} is a coefficient in a function that describes how between-student variation changes with respect to **standlrt** is both valid and useful. This means that in models with complex level 1 variation we do not think of the estimated random parameters as separate variances and covariances. Instead we view them as elements in a function that describes how the level 1 variation changes with respect to explanatory variables. The **Variance function** window can be used to display this function.

- Run the model
- From the **Model** menu, select the **Variance function** window
- From the **level** drop-down list, select **1:student**
- Click **Name**

This produces the following:



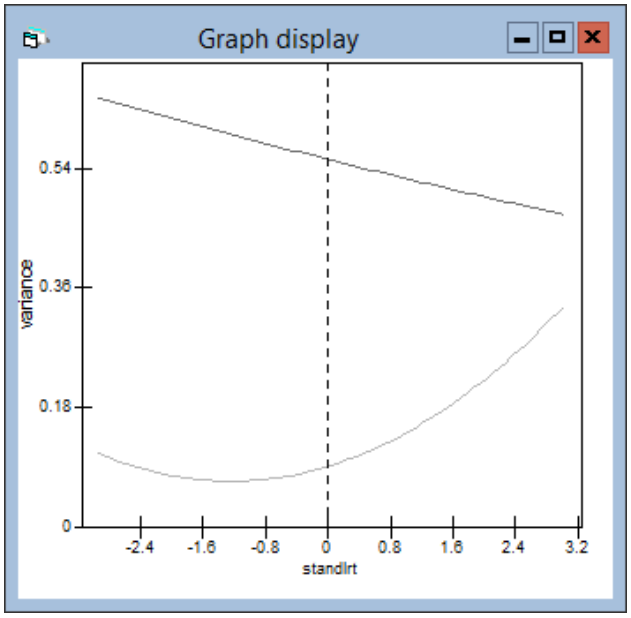
As with level 2, we have a quadratic form for the level 1 variation. Let us evaluate the function for plotting.

- In the **variance output to** drop-down list select **c31**
- Click **calc**

Let's add the level 1 variance function to the graph containing the level 2 variance function.

- Select the **Customised graph** window
- Select the **display** (**ds#**) used to plot the level 2 variance function
- Add another data set with **y** as **c31**, **x** as **standlrt**, plotted as a red line
- Click **Apply**
- To see the level 1 variance function, we need to rescale the **y**-axis to run between **0** and **0.7**
- Also change the **y**-axis label to **variance**

This produces the following plot:



The lower curved line is the between-school variation. The higher straight line is the between-student variation. If we look at the **Equations** window we can see that σ^2_{e1} is zero to three decimal places. The variance σ^2_{e1} acts as the quadratic coefficient in the level 1 variance function; hence we have a straight line. The general picture is that the between-school variation increases as **standlrt** increases, whereas between-student variation decreases with **standlrt**. This means the variance partition coefficient (school variance / [school variance + student variance]) increases with **standlrt**. Therefore the effect of school is relatively greater for students with higher intake achievements.

Notice, as we pointed out earlier, that for high enough levels of **standlrt** the level 1 variance will be negative. In fact, in the present data set such values of **standlrt** do not exist and the straight line is a reasonable approximation over the range of the data.

The student level variance functions are calculated from 4059 points, that is the 4059 students in the data set. The school level variance functions are calculated from only 65 points. This means that there is sufficient data at the student level to support estimation of more complex variance functions than at the school level.

Let's experiment by allowing the student level variance to be a function of gender as well as **standlrt**. We can also remove the σ^2_{e1} term, which we have seen is negligible.

- Add **girl** to the model
- In the **Equations** window click on β_2
- Check the box labelled **i(student)**

The level 1 matrix Ω_e is now a 3×3 matrix.

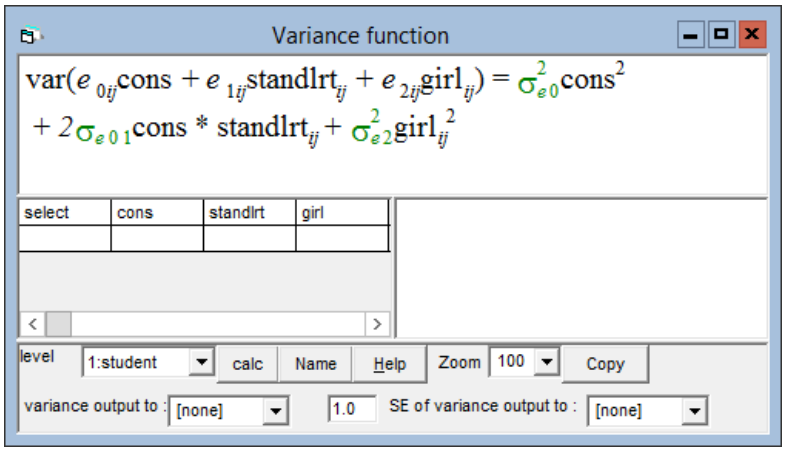
- Click on the σ^2_{e1} term.
- You will be asked if you want to remove the term from the model. Click **Yes**
- Do the same for σ_{e12} and σ_{e02}
- Run the model

When you remove terms from a covariance matrix in the **Equations** window they are replaced with zeros. You can put back removed terms by clicking on the zeros.

Notice that the new level 1 parameter σ^2_{e2} is estimated as -0.054. You might be surprised at seeing a negative variance. Remember, however, that at level 1 the random parameters cannot be interpreted separately; instead they are elements in a function for the variance. What is important is that the function does not go negative within the range of the data.

Note that MLwiN will allow negative values by default for individual variance parameters at level 1. However, at higher levels the default behaviour is to reset any negative variances and all associated covariances to zero. These defaults can be over-ridden in the ***Estimation control*** *window available by pressing the* ***Estimation control*** *on the main toolbar.*

Now use the **Variance function** window to display what function is being fitted to the student level variance.



From the **Equations** window we can see that $\{\sigma^2_{e0}, \sigma_{e01}, \sigma^2_{e2}\} = \{0.583, -0.013, -.054\}$. Substituting these values into the function shown in the **Variance function** window we get the student level variance for the boys:

$0.583 - (0.026 \times \textbf{standlrt})$

For the girls, the variance is:

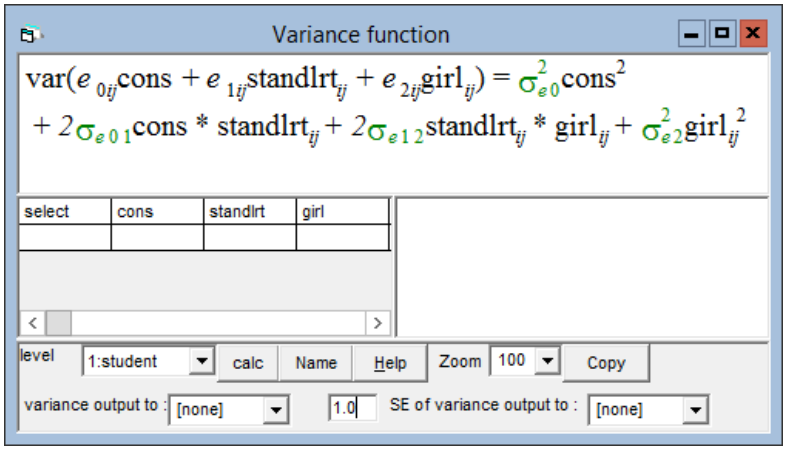
$0.583 - 0.054 - (0.026 \times \mathbf{standlrt})$

> *Note that we can get the mathematically equivalent result by fitting the model with the following terms at level 1: $\sigma^2_{e0}, \sigma_{e01}, \sigma_{e02}$. This is left as an exercise for the reader.*

The line describing the between-student variation for girls is lower than the boys' line by 0.054. It could be that the lines have different slopes. We can see if this is the case by fitting a more complex model for the level 1 variance. In the **Equations** window:

- In the level 1 covariance matrix click on the right hand 0 on the bottom line
- You will be asked if you want to add term **standlrt/girl**. Click **Yes**
- Run the model

We obtain the following estimates for the level 1 parameters $\{\sigma^2_{e0}, \sigma_{e01}, \sigma_{e12}, \sigma^2_{e2}\}$=$\{0.584, -0.034, 0.032, -0.058\}$, and the updated variance function window now looks like this:



The level 1 variance for boys is now:

$$0.584 + (2 \times (-0.034) \times \mathbf{standlrt}) = 0.584 - (0.068 \times \mathbf{standlrt})$$

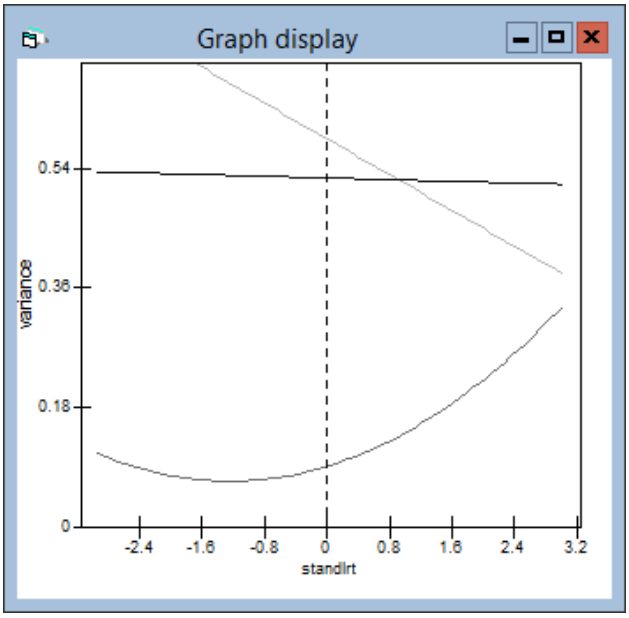
For girls we get:

$$0.584 + (2 \times (-0.034) + 2 \times (0.032)) \times \mathbf{standlrt} - 0.058 = 0.526 - 0.004 \times \mathbf{standlrt}$$

We can see the level 1 variance for girls is fairly constant across **standlrt**. For boys the level 1 variance function has a negative slope, indicating the boys who have high levels of **standlrt** are much less variable in their attainment. We can graph these functions:

- In the **Variance function** window choose **c31** in the **output to:** list
- Click **calc**
- Select the **Customised graphs** window
- Select the data set: **y** = **c31** and **x** = **standlrt**
- In the **group** list select **girl**
- Click **Apply**

This produces the following graph:



We see that the student level variance for boys drops from 0.8 to 0.4 across the spectrum of **standlrt**, whereas the student level variance for girls remains fairly constant at around 0.53.

We are now forming a general picture of the nature of the variability in our model at both the student and school levels of the hierarchy. The variability in schools' contributions to students' progress is greater at extreme values of **standlrt**, particularly positive values. The variability in girls' progress is fairly constant. However, the progress of low intake ability boys is very variable but this variability drops markedly as we move across the intake achievement range.

These complex patterns of variation give rise to intra-school correlations that change as a function of **standlrt** and **gender**. Modelling such intra-unit correlations that change as a function of explanatory variables provides a useful framework when addressing substantive questions.

Fitting models that allow complex patterns of variation at level 1 can produce

useful substantive insights. For example, if from our modelling we know the achievement of some types of student varies considerably, we can infer that amongst this group of students there will be more students at the extremes of achievement. Consequently, the call on resources for special needs will probably be higher where schools have higher proportions of such students.

Also, where there is very strong heterogeneity at level 1, failing to model it can lead to a serious model mis-specification. In some cases the mis-specification can be so severe that the simpler model fails to converge. In such situations, when the model is extended to allow for a complex level 1 variance structure, convergence occurs. Usually the effects of the mis-specification are more subtle; you may find, for example, that failure to model complex level 1 variation can lead to inflated estimates of higher-level variances (that is, between-student heterogeneity becomes incorporated in between-school variance parameters).

Chapter learning outcomes

- ⋆ Use of the *general* notation in MLwiN
- ⋆ That variance functions provide a useful means for interpreting variability at the different levels in our model.
- ⋆ How to construct and graph variance functions in MLwiN
- ⋆ A more complex interpretation of intra-unit correlations

Chapter 8

Getting Started with your Data

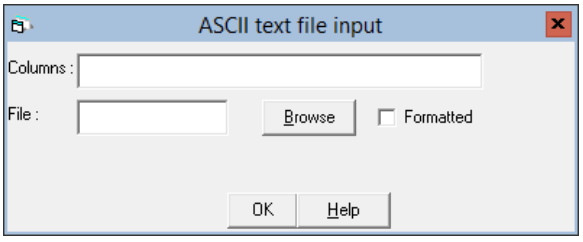
In the previous chapters we have used a prepared example data set. This chapter describes how to get your own data into MLwiN. We also give some advice on commonly experienced problems that occur once you start to fit models to your data.

8.1 Inputting your data set into MLwiN

MLwiN can only input and output numerical data. Data can be input from and output to files or the clipboard. For version 2.02 input/output is from/to text files. Version 2.26 additionally allows input from Stata (`*.dta`), SPSS (`*.sav`), SAS Transport (`*.xpt`), and Minitab (`*.mtw`) files, and also allows data from MLwiN worksheets to be saved in these formats. For documentation see Section 6 of the Manual Supplement for MLwiN Version 2.26.

Reading in an ASCII text data file

If you have data prepared in ASCII format, you may use the **ASCII text file Input** option from the **File** menu to input them. Clicking on this option brings up the following window:



To read in your data set do the following:

- In the **Columns** box type the column numbers into which the data are to be read. If columns are consecutively numbered, you can refer to the range of columns by typing the first and last separated by a '-', e.g. **c2-c6**.
- In the **File** box, you can either type the full path and name for the data file or click the **Browse** button to display the folder structure and allow you to make a selection. (Note that file names are not restricted to having particular extensions such as .txt.)
- If the data are delimited by spaces, commas or tabs, and if each record contains the same number of variables, then clicking on **OK** will read the data into the specified columns.

If the data in each record have the same format, i.e, fixed width variables, you will probably want to specify the input format to MLwiN. Doing so is particularly useful in order to skip certain fields that will not be needed in a modelling session. Checking the **Formatted** box opens up a **Format** box into which you type the data format — a string of comma-separated integers.

*Note that you do not tell MLwiN in the **Format** box about the number of decimal places used for each variable.*

MLwiN recognises two formatting codes: x (a positive integer) means "read a variable of width x characters", and $-y$ means "skip (ignore) y characters". So, for example, to skip the first character in the file, read two 3-digit numbers, skip two characters, and read a 1-digit number, you would type the following in the box: $-1, 3, 3, -2, 1$

Writing data to a specified text file operates in a similar way through the **ASCII text file output** option. If the data are not formatted, each case's values for the different variables are separated by spaces in the output file.

If any data item in a data column contains non-numeric characters, then that data column will be converted to a categorical variable. If a column contains a mixture of data items where some items are numbers and other items are repeated instances of a single textual pattern, containing non-numeric characters (e.g. *), then that textual pattern is treated as a code for missing data.

Common problems that can occur in reading ASCII data from a text file

Some common problems that can occur with the inputting of ASCII text files are:

- The data file includes a list of column names at the top; some packages save the column names to the top of the data file when using the Save option.
- The data file contains missing values that were converted to either blank spaces or illegal characters when the file was saved in another package.
- The data file uses ',' rather than '.' to represent a decimal point. Although MLwiN will display worksheets using whichever representation is set on a computer, when inputting data from another package, the '.' must be used.
- The number of columns given in the **ASCII text file input** window's **Columns** box does not correspond to the number of columns present in the input file.

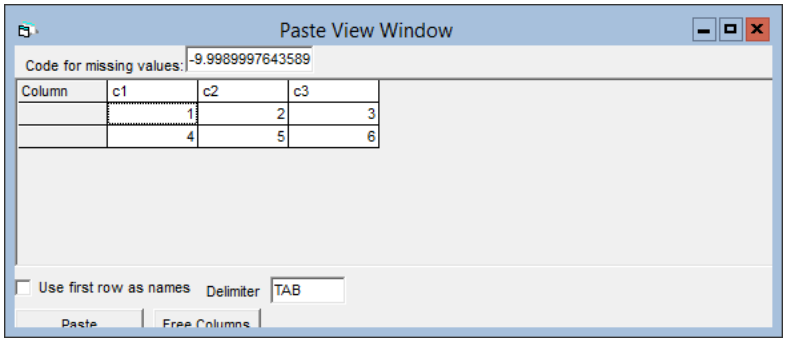
All of these sources of data input errors can be checked by viewing the data set using the software package that the data are being exported from, or by looking at the data file with a word processor. To correct the data, remove any headers containing variable names or other information and use the software's Find and Replace feature to globally convert any illegal characters. If the data contain missing values, it is sensible to convert these to a unique value, for example -999 that can then be exploited after the data have been successfully input into MLwiN. We will say more about missing values later in this chapter.

If you have a very large data set, make sure that you have specified a large enough worksheet size using the **Settings** window — accessed by selecting the **Worksheet** option on the **Options** menu. It is also a good idea to input a large data set in several stages, i.e, reading a subset of the variables each time (into consecutive sets of worksheet columns).

Pasting data into a worksheet from the clipboard

If you have your data in another package such as EXCEL or SPSS it may often be more convenient to copy your data from these packages and then paste them into MLwiN. If you have copied data onto the clipboard from another application then they can be pasted into MLwiN through the **Paste** option on the **Edit** menu[1]. For example, if we have a 2 by 3 table of numbers in the clipboard and we select Paste, the following window appears:

[1]Or using the **Paste** button on the **Names** window; for documentation see Section 8.2.4 of the Manual Supplement for MLwiN Version 2.10



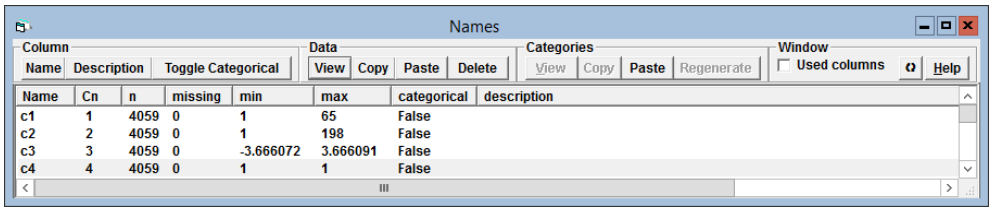
This window allows you to view the data and assign it to MLwiN columns. You can select the next free columns on the worksheet by pressing the **Free Columns** button. You can also choose which MLwiN columns the data are to be assigned to by clicking in the top row of the data table and selecting MLwiN columns from the list that appears.

If the first row of your pasted data contains variable names, then checking the **Use first row as names** box will assign these names to the MLwiN copies of the variables.

As in the case of reading ASCII data from a file, if you have a very large data set, make sure that you have specified a large enough worksheet size before you start pasting in the data. It is also a good idea to paste data in stages (into consecutive sets of columns).

Naming columns

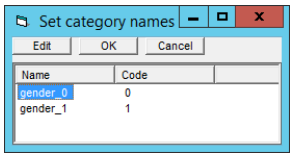
When using the **ASCII text file input** option, MLwiN does not allow the user to input the column names directly from the file into a worksheet. Columns can instead be named using the **Names** window, which is accessed via the **Data Manipulation** menu. The figure below shows an example of a data set as it appears in the **Names** window before variable names have been assigned to the worksheet columns.



A column to be (re)named is selected by clicking on its column number (c1, c2, c3 etc.) in the column headed **Name**. The selected column's (current) name is displayed in the text box where it can be edited to specify any desired name. After editing, pressing **return** updates the column's name.

Adding category names

When a variable is categorical, names can also be given to the individual categories. This is useful because it allows MLwiN to create and name dummy variables for analysis, and to annotate tables etc. Suppose column **c4** in the above data set has been named **gender**. We can declare **gender** to be a categorical variable by selecting **gender** in the **Names** window and clicking the **Toggle Categories** button. We can then name the categories by pressing the **View** button in the categories section. This produces the following window:



Clicking in the **name** column and typing text beside each category value allows category names to be assigned.

Missing data

MLwiN assigns a single code value for any missing data. The default value is a large negative number (−9.999E+29) called the *system missing* value. The numerical value to represent missing data can be set by the user in the bottom box on the **Numbers** tab of the **Settings** window. (This window is accessed from the **Options** menu by choosing **Worksheet**.) You may have input data containing a specific missing value code, −99, say. In this case you should set MLwiN's missing data value to −99.

Note that before inputting the data to MLwiN, it is helpful to check that you have used the same missing value code for every variable and that the code you have chosen is not a legitimate value for any of your variables.

When a user specifies a missing value code such as −99, all occurrences of that value in the data set are changed to MLwiN's system missing value. If the value you specified as a missing value code is a legitimate value for some of your variables, MLwiN will not make the distinction. There is a 1-step recovery if you change your mind about your choice of missing value code. If you reset the missing code, you will be prompted to see whether you wish your previous code to revert to its original value. If not, then the old *and* new codes are treated as missing.

*Note that you can also use the **Recode variables** window to change a missing value to another code. (See HELP for instructions on doing*

this.) Note also that the ***Calculate*** *window allows the missing code to be used in logical expressions.*

It is important to understand that missing data are automatically ignored in model fitting and that *a likelihood ratio statistic* comparing two models with different amounts of missing data is not valid. The **Equations** window reports how many cases were used in each model.

Unit identification columns

MLwiN holds numbers in single precision; this allows 6 digits of precision. Sometimes unit identification columns contain very long numbers, e.g, 10000001, 10000002 etc. Since these numbers (in this example) vary only in the 8th digit, they will be indistinguishable to MLwiN. Normally, if long numbers such as these were to be used in arithmetic calculations, the indistinguishability would not be a problem. However, if the numbers are used to denote different units, e.g, schools, then there is a problem. When you import data and MLwiN encounters a variable whose values have more than 6 digits of precision, you will be offered the option of converting the variable to a categorical variable. This means that the numbers read in are treated as category labels and each distinct label is given an integer number from 1 to m, where m is the number of distinct labels.

Saving the worksheet

Once you have input and named your data, you should save your data as an MLwiN worksheet using the **Save worksheet** option on the **File** menu. While working with MLwiN it is well worth saving your worksheet at regular intervals as a backup. (When you fit a series of different models to the same data, you may want to save each step's work in a different worksheet using the **Save worksheet As** option.)

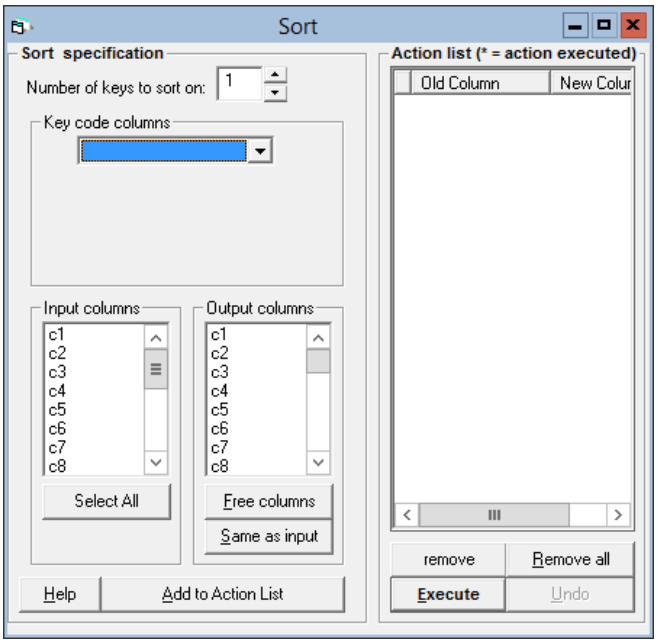
Sorting your data set

The most common mistake new users make when trying to fit a multilevel model to their data set is that they do not sort the data set to reflect the data's hierarchical or nested structure. (This is an easy mistake to make.) All the examples in this manual have already been sorted into the correct structure — students within schools, in the case of the data set used in the previous chapters.

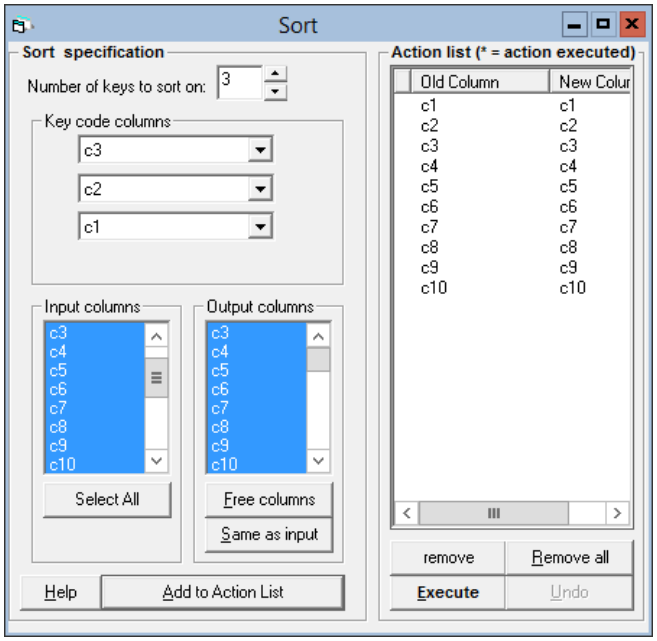
The data must be sorted so that all records for the same highest-level unit are grouped together and within this group, all records for a particular lower level unit are contiguous. For example the following represents the first few records of a sorted three-level data set:

Level 3 ID	Level 2 ID	Level 1 ID
1	1	1
1	1	2
1	1	3
1	2	1
1	2	2
1	2	3
2	1	1
2	1	2

The **Sort** window (see below), accessed via the **Data Manipulation** menu, can be used to reorder data records.



We assume that the level 1, 2 and 3 identifiers of our model are stored in columns **c1**, **c2** and **c3**, respectively and that the response and all predictor variables can be found in columns **c4** to **c10**. To sort this data structure correctly, we need to set the following on the **Sort** window as shown here:



A full explanation of how to use both the **Sort** window and other data manipulation windows can be found in the on-line **Help** system.

> *Note that all of the columns to be sorted (c1 to c10 in this illustration) must be of the same length and must contain data for the sort operation to work correctly.*

Many users ask why the software cannot sort the data itself, and there are several reasons for this. Firstly the software doesn't know which columns the user wants to sort. Secondly because of choices made in assigning unit identification codes within the data set, it may not be possible to *automatically* take columns of data that are appropriately sorted for fitting a particular model and perform an unambiguous re-sorting to create a hierarchy suitable for fitting a different model. To see this, consider the above table of sorted data and suppose that instead of a fitting a 3 level model we wanted to drop level 3. We would then have several student records that have the same level 2 ID (1, in this case), but which do not actually belong to the same level 2 unit. In an educational scenario they could be from class 1 in school 1 and from class 1 in school 2, i.e, distinct classes.

Once you have sorted your data and set up a model, you should check the **Hierarchy viewer** (accessed from the **Model** menu) to ensure that the data structure that the software reports — in terms of number of units at each level — is as you expect.

8.2 Fitting models in MLwiN

Once you have input the data into MLwiN named the columns and saved the worksheet, it is often tempting to go straight ahead and fit a really complicated model with lots of fixed and random effects. Then you may well come across several problems, for example the model does not converge, has numerical problems or gives unexpected answers. The main piece of advice here is that multilevel modelling is like any other type of statistical modelling and a useful strategy is to start by fitting simple models and slowly increase the complexity. In the rest of this section we will list some of the main pointers that should be followed to reduce frustration while trying to fit multilevel models in MLwiN.

What are you trying to model?

It is important before starting to fit models to your data set to know as much as possible about your data and to establish what questions you are trying to answer. It is important to identify which variable(s) are your response variable(s) of interest. It is also important to establish, particularly if you are new to multilevel modelling, what is meant by the terms: levels, predictors, fixed effects and random effects, and to identify which variables in your data set contain ID codes for units, i.e. represent levels, and which are measured variables. If you are not sure what these terms mean, then you need to work through Chapters 1 to 6 of this manual before proceeding with your own data.

Do you really need to fit a multilevel model?

It is always a good idea to do some more basic statistical analysis before proceeding on to multilevel modelling. Plotting the response variable against several predictors will allow you to examine graphically whether there are any strong relationships. Fitting simple single level models before proceeding to multilevel models is also a good idea, particularly as the fixed effects estimates from a single level model should generally be similar to those achieved by a corresponding multilevel model.

One point to note is that just because a model has more levels, more fixed effects and more random effects this does not automatically mean that it will be a better model. Often the opposite is true. A distinction should be made here between trying to fit a multilevel model to a data set that is too small and to a data set where there is no higher-level variation. A data set that only has 4 level 2 units is best fitted as a single level model with the level 2 units included as 3 dummy variables. Fitting a multilevel model to this

data will almost certainly report no level 2 variation. However, this is not a generalisable statement; we simply have not sampled enough level 2 units.

Have you built up your model from a variance components model?

A sensible way of fitting a multilevel model is to start with the basic variance components model (as in the tutorial example). Then you can build up models of increasing complexity by adding predictors that are deemed to be important and checking whether they have substantial and / or significant fixed or random coefficients. If instead you add lots of predictors into your model and have convergence problems, it may be difficult to establish which predictor is causing the problem. Building up a model by adding variables one at a time and using the MORE option rather than START also has less chance of producing convergence problems with the IGLS and RIGLS estimation methods. This is because the estimates from the last model fitted are used as starting values for the new model.

Have you centred your predictor variables?

If your data set contains continuous predictor variables, there are several benefits to be gained from centring them, i.e, subtracting a variable's mean from each case's value of that variable. The primary benefit in so doing is that it often makes interpretation of the intercept term in the model easier, as it is now the predicted value for a subject that has average values for each explanatory variable. This is generally more useful than the response value for a subject with zero for all predictors, because zero may not be a typical value for the corresponding explanatory variable. Centring predictor variables can also reduce the chances of numerical errors in the IGLS and RIGLS estimation methods and reduce the correlation in the chains produced by MCMC methods.

Chapter learning outcomes

- ⋆ How to input data into MLwiN
- ⋆ How to sort data and set missing values
- ⋆ How to set up categorical variables
- ⋆ How to avoid some of the common mistakes users can make when modelling data

Chapter 9

Logistic Models for Binary and Binomial Responses

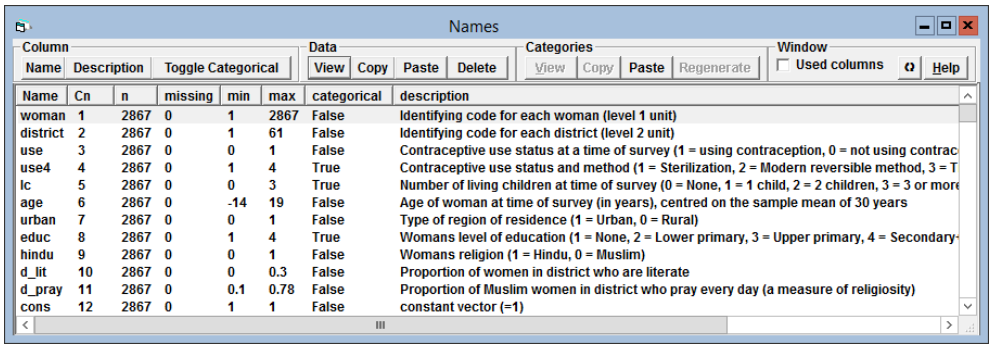
9.1 Introduction and description of the example data

So far, we have considered multilevel models for continuous response variables. In this chapter, we look at models for binary or binomial (proportion) responses. We begin with a discussion of single-level models for binary responses, focusing on the popular logit model but giving a brief discussion of other link functions such as the probit. We then show how the single-level model can be extended to handle data with a two-level hierarchical structure, leading to a two-level random intercepts logistic model. Significance testing and model interpretation using odds ratios and variance partition coefficients are discussed. Next we consider a random coefficient (slope) model for binary data. Finally, we illustrate how logistic models can be fitted when the response is a proportion (i.e, binomial) rather than binary and discuss models that allow for extra-binomial variation.

The data for the examples in this chapter are a sub-sample from the 1989 Bangladesh Fertility Survey (Huq & Cleland, 1990). The binary response variable that we consider refers to whether a woman was using contraception at the time of the survey. The full sample was analysed in Amin et al. (1997), but with a multinomial response that distinguished between different types of contraceptive method. In Chapter 10 we will consider the same multinomial response. The aim of the analysis in this chapter is to identify the factors associated with use of contraception and to examine the extent of between-district variation in contraceptive use. The data have a two-level hierarchical structure, with 2867 women nested within 60 districts.

We will begin by opening the MLwiN worksheet **bang.ws** using the **Open**

Worksheet option from the **File** menu. The following **Names** window will be displayed:



Name	Cn	n	missing	min	max	categorical	description
woman	1	2867	0	1	2867	False	Identifying code for each woman (level 1 unit)
district	2	2867	0	1	61	False	Identifying code for each district (level 2 unit)
use	3	2867	0	0	1	False	Contraceptive use status at a time of survey (1 = using contraception, 0 = not using contrac
use4	4	2867	0	1	4	True	Contraceptive use status and method (1 = Sterilization, 2 = Modern reversible method, 3 = T
lc	5	2867	0	0	3	True	Number of living children at time of survey (0 = None, 1 = 1 child, 2 = 2 children, 3 = 3 or more
age	6	2867	0	-14	19	False	Age of woman at time of survey (in years), centred on the sample mean of 30 years
urban	7	2867	0	0	1	False	Type of region of residence (1 = Urban, 0 = Rural)
educ	8	2867	0	1	4	True	Womans level of education (1 = None, 2 = Lower primary, 3 = Upper primary, 4 = Secondary+
hindu	9	2867	0	0	1	False	Womans religion (1 = Hindu, 0 = Muslim)
d_lit	10	2867	0	0	0.3	False	Proportion of women in district who are literate
d_pray	11	2867	0	0.1	0.78	False	Proportion of Muslim women in district who pray every day (a measure of religiosity)
cons	12	2867	0	1	1	False	constant vector (=1)

The variables are defined as follows:

Variable	*Description*
woman	Identifying code for each woman (level 1 unit)
district	Identifying code for each district (level 2 unit)
use	**Contraceptive use status at time of survey**
1	using contraception
0	not using contraception
use4	**Contraceptive use status and method**
1	Sterilization (male or female)
2	Modern reversible method
3	Traditional method
4	Not using contraception
lc	**Number of living children at time of survey**
0	None
1	1 child
2	2 children
3	3 or more children
age	Age of woman at time of survey (in years), centred on the sample mean of 30 years
urban	**Type of region of residence**
1	Urban
0	Rural
educ	**Woman's level of education**
1	None
2	Lower primary
3	Upper primary
4	Secondary+
hindu	**Woman's religion**
1	Hindu
0	Muslim
d_lit	Proportion of women in district who are literate
d_pray	Proportion of Muslim women in district who pray every day (a measure of religiosity)
cons	constant vector

In this chapter we will analyse the binary response **use**. The multinomial response **use4** will be analysed in Chapter 10.

9.2 Single-level logistic regression

Link functions

We will begin by fitting a single-level logistic regression model with a single explanatory variable x_i. The binary (0,1) response for the ith unit (here, woman) is denoted by y_i. We denote the probability that $y_i = 1$ by π_i. A general model for binary response data is:

$$f(\pi_i) = \beta_0 + \beta_1 x_i$$

where $f(\pi_i)$ is some transformation of π_i, called the *link function*. Popular choices for the link function are:

The *logit* link, i.e, $f(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right)$, where the quantity $\pi_i/(1-\pi_i)$ is the *odds* that $y_i = 1$.

The *probit* link, where $f(\pi_i) = \Phi(\pi_i)$ is the cumulative density function of the standard Normal distribution.

The *complementary log-log* link, i.e, $f(\pi_i) = \log(-\log(1-\pi_i))$. We will call this the clog-log link, but it is sometimes referred to as the log-log link.

All of the above transformations ensure that predicted probabilities $\hat{\pi}$ derived from the fitted model will lie between 0 and 1. In practice, the significance of coefficients and predictions of π are fairly robust to the choice of link function. The logit transformation tends to be most widely used[1], mainly because the exponentiated coefficients from a logit model can be interpreted as odds ratios. For this reason, we will focus on logit models in this chapter, although we will show how other link functions can be fitted in MLwiN.

The logit model takes the form:

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_i \tag{9.1}$$

[1]The probit transformation is also popular, particularly in economics. See Collett (1991) for a comparison of the logit, probit and clog-log link functions.

Interpretation of coefficients

Taking exponentials of each side of (9.1), we obtain:

$$\frac{\pi_i}{1-\pi_i} = e^{\beta_0} \times e^{\beta_1 x_i} \tag{9.2}$$

If we increase x by 1 unit, we obtain:

$$\frac{\pi_i}{1-\pi_i} = e^{\beta_0} \times e^{\beta_1 (x_i+1)} = e^{\beta_0} \times e^{\beta_1 x_i} \times e^{\beta_1}$$

This is the expression in (9.2) multiplied by e^{β_1}. Therefore e^{β_1} can be interpreted as the multiplicative effect on the odds for a 1-unit increase in x. If x is binary (0,1), then e^{β_1} is interpreted as the *odds ratio*, comparing the odds for units with $x = 1$ relative to the odds for units with $x = 0$.

If we rearrange (9.2), we obtain an expression for π_i:

$$\pi_i = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)} = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_i))} \tag{9.3}$$

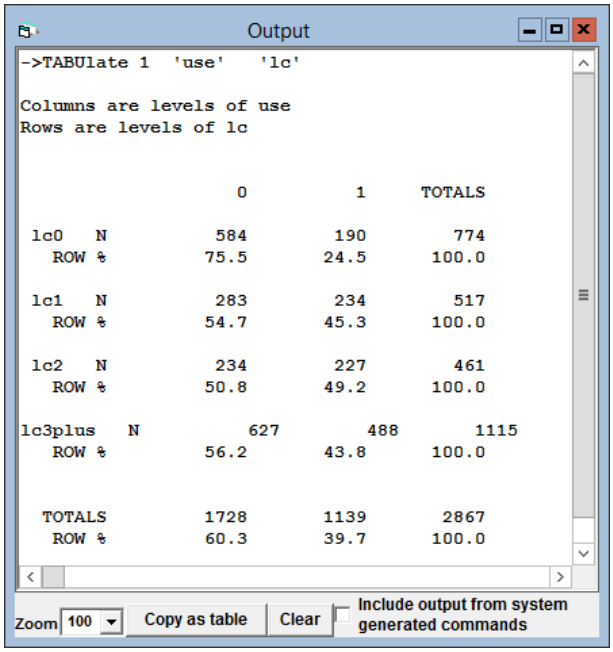
One way of interpreting a fitted model is to compute predicted probabilities for a range of values of x, substituting the estimates of β_0 and β_1 in (9.3).

Fitting a single-level logit model in MLwiN

We will begin by examining the relationship between contraceptive use (**use**) and number of living children (**lc**). Before carrying out a logistic regression analysis, we can examine a tabulation of the percentage using contraception (and not using contraception) by number of children:

- From the **Basic Statistics** menu, select **Tabulate**
- Next to **Columns**, select **use** from the pull-down list
- Check **Rows** and select **lc** from the pull-down list
- Under **Display**, check **Percentages of row totals**
- Click **Tabulate**

You should obtain the following table of percentages:


Output
->TABUlate 1 'use' 'lc'
Columns are levels of use
Rows are levels of lc
0 1 TOTALS
lc0 N 584 190 774
ROW % 75.5 24.5 100.0
lc1 N 283 234 517
ROW % 54.7 45.3 100.0
lc2 N 234 227 461
ROW % 50.8 49.2 100.0
lc3plus N 627 488 1115
ROW % 56.2 43.8 100.0
TOTALS 1728 1139 2867
ROW % 60.3 39.7 100.0
Zoom 100
Copy as table
Clear
Include output from system generated commands

From this table, we see that the percentage using contraception is markedly lower for women with no children compared to women with one or more children.

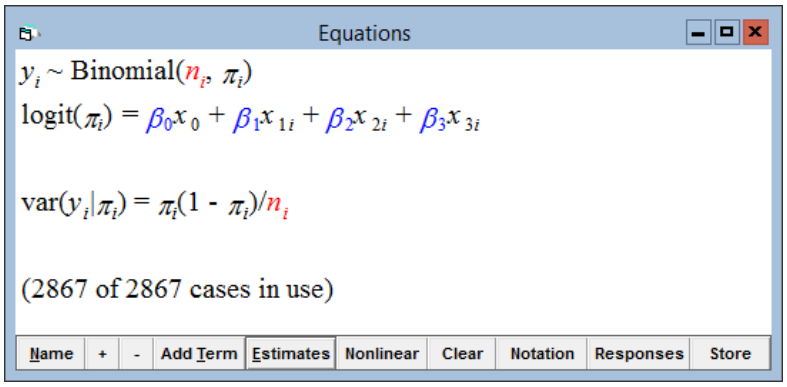
We will now model this relationship by fitting a single-level model to the binary response variable use and including dummy variables for **lc** as explanatory variables.

- From the **Model** menu, select **Equations**
- Click on the **Name** button
- Click on ***y*** and in the **Y variable** window, make the following selections:
  y: **use**
  N levels: **1-i**
  level 1(i): **woman**
- Click **done**
- Click on the **N** in the **Equations** window
- In the **Response type** window, select **Binomial** as the distribution and **logit** as the link function.
- Click **Done**
- Click on x_0, select **cons** from the drop-down list presented and click **Done**
- Click on the **Add term** button, and from the **variable** drop-down list, select **lc**.
- Click **Done**
- Click **Estimates**

Note that the default link function is the logit, but notice that the probit and clog-log links are other options.

Note that when this example worksheet was prepared ***lc*** *was declared to be a categorical variable. Therefore MLwiN automatically enters dummy variables when* ***lc*** *is selected as an explanatory variable. By default, the first category (****lc0****) which corresponds to 'no children' is taken as the reference.*

The **Equations** window should look like this:



Since **lc** has four categories, three dummy variables have been added to the model.

The first line in the **Equations** window states that the response variable follows a binomial distribution with parameters n_i and π_i. The parameter n_i is known as the denominator. In the case of binary data, n_i is equal to 1 for all units. We will now create n_i and call the new variable **denom**.

- From the **Data Manipulation** menu, select **Generate vector**
- In the **Generate Vector** window, for **Output column**, select **c17**
- For **Number of copies**, type **2867**
- For **Value**, type **1**
- Click **Generate**, and close the window
- From the **Data Manipulation** menu, select **Names**
- In the **Name** column of the **Names** window, select **c17**
- Click the **Edit name** button, type **denom** and press return
- Now in the **Equations** window, click on n_i
- In the **specify denominator** window, select **denom** from the drop-down list
- Click **Done**

Note that if our data had been binomial (i.e, in the form of proportions) then n_i would be equal to the number of units on which the proprtion is based. For example, if π_i was the proportion of women who used contraception in district i then n_i would be the number of women of reproductive age in district i.

The second line in the **Equations** window is the equation for the logit model, which has the same form as (9.1) since $x_0 = 1$ for all women. (This is the **cons** variable created by MLwiN.) If you click on the **Name** button, you will see the variable names.

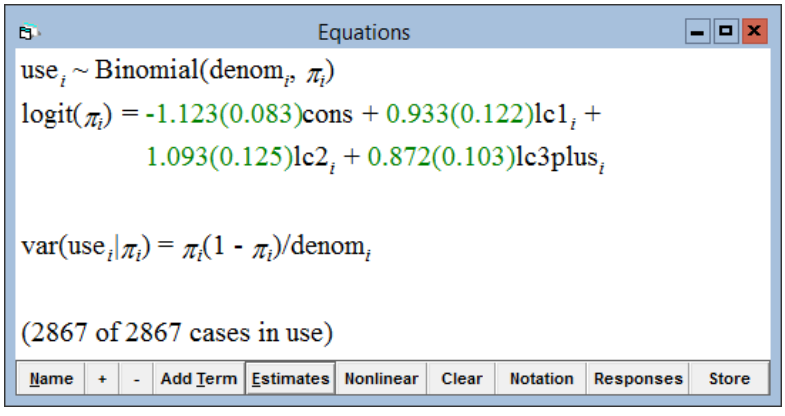
Before fitting the model, we have to specify details about the estimation procedure to be used. The estimation choices will be discussed when we come to fit multilevel models.

- Click on the **Nonlinear** button at the bottom of the **Equations** window
- In the **Nonlinear estimation** window, click on **Use Defaults**, then **Done**

Now to fit this model:

- Click **Start**

After clicking on **Estimates** twice you should see the following:



The last line in the **Equations** window states that the variance of the binomial response is $\pi_i(1-\pi_i)/\mathbf{denom}_i$, which in the case of binary data, simplifies to $\pi_i(1-\pi_i)$.

The variables **lc1**, **lc2** and **lc3plus** are indicators for '1 child', '2 children', and '3+ children' respectively. The fitted model has the equation:

$$\text{logit}(\pi_i) = -1.123 + 0.933\mathbf{lc1}_i + 1.093\mathbf{lc2}_i + 0.872\mathbf{lc3plus}_i$$

We can calculate odds ratios, comparing the categories coded 1, 2 or 3 with the category coded 0, simply by taking exponentials of the coefficients of **lc1**, **lc2** and **lc3plus**. Also shown are Z-ratios, which can be compared with a standard Normal distribution to carry out pairwise tests of differences between categories 1, 2 or 3 and category 0.

Category of **lc**	β	S.E.	Z=β/SE	e^{β}
None	0	-	-	1
1	0.933	0.122	7.65	2.54
2	1.093	0.125	8.74	2.98
3+	0.872	0.103	8.47	2.39

Women with children have a significantly higher odds (or probability) of using contraception than women without children. The odds of using contraception increases with number of children (with a slight decrease for 3 or more), but the largest shift is between 0 and 1 children. The odds of using contraception for a woman with one child are 2.54 times the odds for a woman with no children. We could also calculate odds ratios comparing pairs of **lc** categories that do not involve the reference category. For example, the odds of using contraception for a woman with two children are $2.98/2.54 = 1.17$ times the odds for a woman with one child.

We can also use the estimated coefficients to calculate predicted probabilities of contraceptive use for each category of lc. For example, using equation (9.3), the probability of using contraception for a woman with one child is estimated as:

$$\hat{\pi} = \frac{1}{1 + \exp(-(-1.123 + 0.933))} = 0.45$$

The predicted probabilities for each category of **lc** are given below.

Category of **lc**	$\hat{\pi}$
None	0.245
1	0.453
2	0.492
3+	0.438

Notice that the predicted probabilities of using contraception agree with the sample proportions listed in the table of percentages shown earlier.

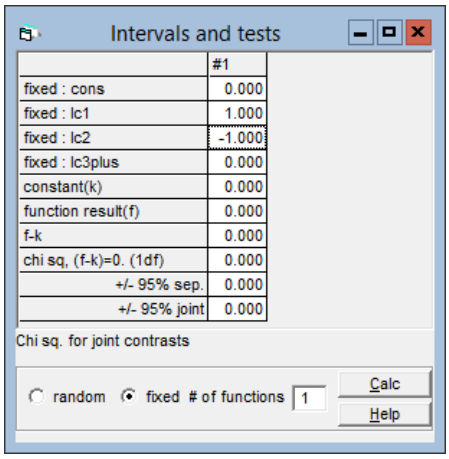
Since the estimated coefficients for **lc** categories 1 and 2 are fairly similar, we might want to test whether there is a difference between these categories in

the probability of using contraception.[2] We can carry out a Wald test to test the null hypothesis that $\beta_1 = \beta_2$ (where β_1 and β_2 are the coefficients of **lc1** and **lc2** respectively). The null hypothesis can also be written $\beta_1 - \beta_2 = 0$, or in matrix form as

$$\begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = 0$$

To carry out the test in MLwiN:

- From the **Model** menu, select **Intervals and tests**
- At the bottom of the **Intervals and tests** window, click the radio button next to **fixed**
- Next to **# of functions**, we retain the default of **1**
- Edit the values next to **lc1** and **lc2** in **column #1** as shown below



After clicking **Calc**, you should obtain a test statistic (**joint chi sq test(1df)**, which appears where before you pressed **Calc** it said **Chi sq. for joint contrasts**) of 1.548 on 1 d.f. We can compute a p-value as follows:

- From the **Basic Statistics** menu, select **Tail Areas**
- Next to **Value**, type **1.548**
- Next to **Degrees of freedom**, type **1**
- Click **Calc**

The p-value is 0.213, so we conclude that the difference in the probability of using contraception between women with 1 child and women with 2 children is not significant at the 5% level. We would therefore be justified in simplifying the model by collapsing categories 1 and 2 of **lc**, but we will retain the existing categories here.

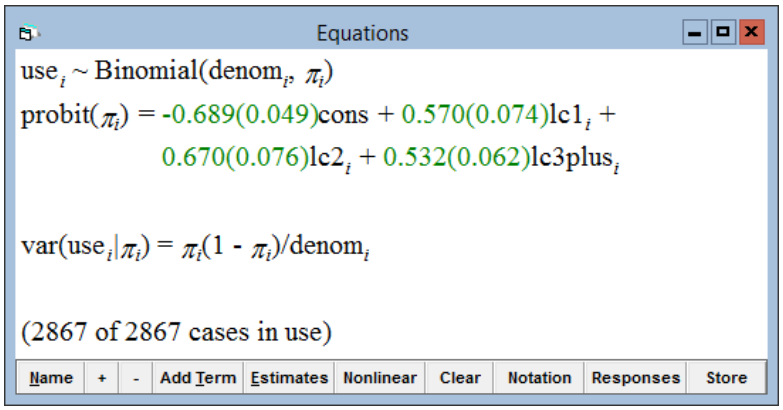
[2]The estimate for **lc3plus** is also close to the estimates for **lc1** and **lc2**, so we might wish to test for a difference between all three categories. For our illustration, however, we will restrict the comparison to categories 1 and 2.

A probit model

We can fit a probit model with the same explanatory variables, simply by changing the link function in the **Equations** window from logit to probit.

- Click on **logit** in the **Equations** window
- In the **Response type** window, under **Select link function**, check **probit**
- Click **Done**
- Click on **Start** to fit this model

You should see the following results:


Equations
use$_i$ ~ Binomial(denom$_i$, π_i)
probit(π_i) = -0.689(0.049)cons + 0.570(0.074)lc1$_i$ +
0.670(0.076)lc2$_i$ + 0.532(0.062)lc3plus$_i$
var(use$_i$|π_i) = π_i(1 - π_i)/denom$_i$
(2867 of 2867 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

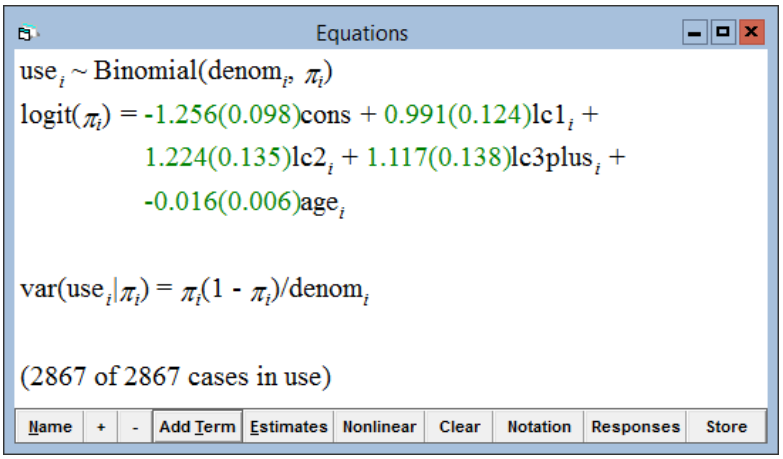
Notice that although the magnitudes of the coefficients have changed, they are in the same direction as in the logit model. The pattern in the effect of **lc** is also the same as in the logit model. If you were to calculate Z statistics, you would find that these are also very close to those obtained from the logit model. We will thus consider only logit models from now on.

- Click on **probit** in the **Equations** window, and under **Select link function** check **logit**
- Click **Done**
- Click **Start** to re-fit the logit model

We will also add a further explanatory variable, age, to the model.

- In the **Equations** window, click **Add term**
- In the **Specify term** window, select **age** from the **variable** drop-down list
- Click **Done**
- Click **More** to fit the new model

The results are as follows:


Equations
use_i ~ Binomial(denom_i, π_i)
logit(π_i) = -1.256(0.098)cons + 0.991(0.124)lc1_i +
1.224(0.135)lc2_i + 1.117(0.138)lc3plus_i +
-0.016(0.006)age_i
var(use_i|π_i) = π_i(1 - π_i)/denom_i
(2867 of 2867 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

We can see that the probability of using contraception decreases with **age**, adjusting for the effect of number of children. This effect is statistically significant at the 5% level; we leave you to carry out the Wald test as an exercise.

9.3 A two-level random intercept model

Model specification

We will now extend our model to allow for district effects on the probability of using contraception. We begin with a random intercept or variance components model that allows the overall probability of contraceptive use to vary across districts. Our binary response is y_{ij} which equals 1 if woman i in district j was using contraception, and 0 if she was not. Similarly, a j subscript is added to the proportion so that $\pi_{ij} = \Pr(y_{ij} = 1)$. If we have a single explanatory variable, x_{ij}, measured at the woman level, then (9.1) is extended to a two-level random intercept model as follows:

$$\begin{aligned} \text{logit}(\pi_{ij}) &= \beta_{0j} + \beta_1 x_{ij} \\ \beta_{0j} &= \beta_0 + u_{0j} \end{aligned} \tag{9.4}$$

As in a random intercept model for a continuous response, the intercept consists of two terms: a fixed component β_0 and a district-specific component, the random effect u_{0j}. As before, we assume that the u_{0j} follow a Normal distribution with mean zero and variance σ^2_{u0}.

Estimation procedures

For discrete response multilevel models, maximum likelihood estimation is computationally intensive, and therefore quasi-likelihood methods are implemented in MLwiN. These procedures use a linearisation method, based on a Taylor series expansion, which transforms a discrete response model to a continuous response model. After applying the linearisation, the model is then estimated using iterative generalised least squares (IGLS) or reweighted IGLS (RIGLS). See Goldstein (2003) for further details. The transformation to a linear model requires an approximation to be used. The types of approximation available in MLwiN are: marginal quasi-likelihood (MQL) and predictive (or penalized) quasi-likelihood (PQL). Both of these methods can include either 1st order terms or up to 2nd order terms of the Taylor series expansion. The 1st order MQL procedure offers the crudest approximation and may lead to estimates that are biased downwards, particularly if sample sizes within level 2 units are small or the response proportion is extreme. An improved approximation procedure is 2nd order PQL, but this method is less stable and convergence problems may be encountered. It is for this reason that in the analysis below we begin with the 1st order MQL procedure to obtain starting values for the 2nd order PQL procedure. Intermediate choices, 1st order PQL and 2nd order MQL, are also often useful. Further details of these quasi-likelihood procedures can be found in Goldstein (2003).

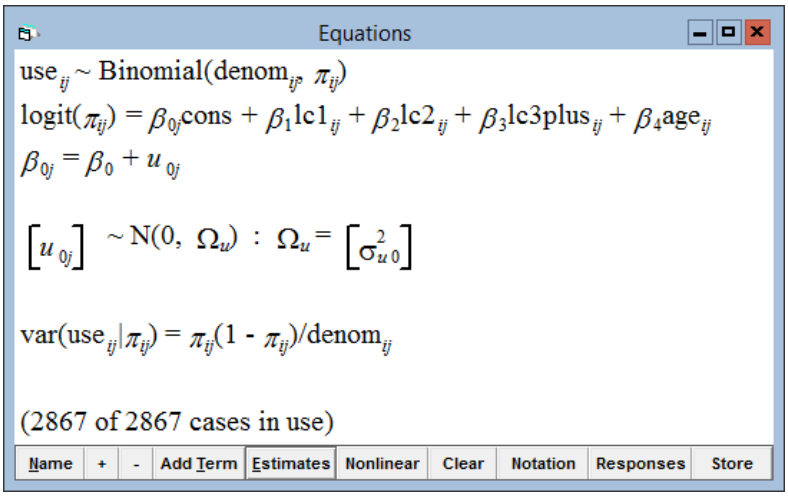
An alternative to likelihood-based estimation procedures is to use a Monte Carlo Markov Chain (MCMC) method, also implemented in MLwiN. In *MCMC Estimation in MLwiN* (Browne, 2003), there is a tutorial in which these Bangladesh contraceptive use data are reanalysed using MCMC methods.

Fitting a two-level random intercept model in MLwiN

We will now extend the model fitted at the end of Section 9.2 to a random intercept model. We first need to declare that the data have a two-level hierarchical structure, with district at the higher level, and then allow the intercept β_0 to vary randomly across districts.

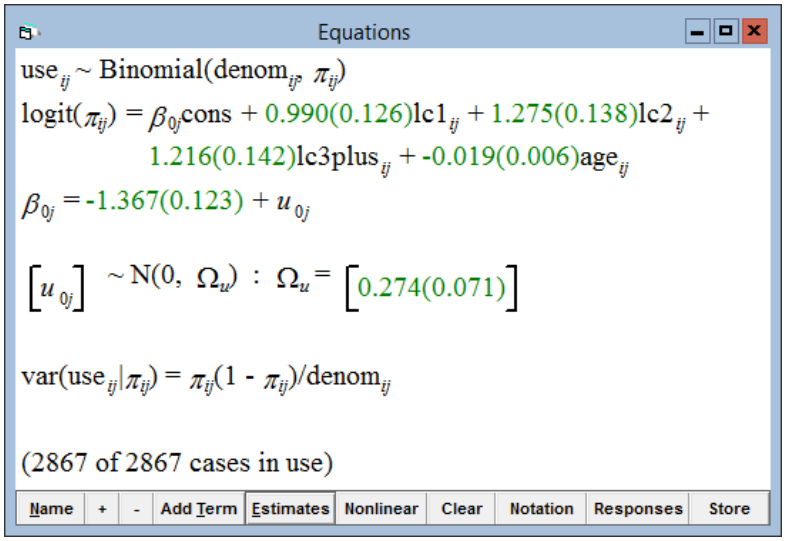
- Click on **use**$_i$ to open the **Y variable** window
- Change **N levels:** from **1-i** to **2-ij**
- Next to **level 2(j):** select **district** from the drop-down list
- Click **Done**
- Now click on **cons** (or its coefficient β_0) in the **Equations** window
- Check **j(district)** in the **X variable** window
- Click **Done**

You should see the following window (you may need to click on **Estimates** first):


Equations
use$_{ij}$ ~ Binomial(denom$_{ij}$, π_{ij})
logit(π_{ij}) = β_{0j}cons + $\beta_1$lc1$_{ij}$ + $\beta_2$lc2$_{ij}$ + $\beta_3$lc3plus$_{ij}$ + $\beta_4$age$_{ij}$
β_{0j} = β_0 + u_{0j}
$\begin{bmatrix} u_{0j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} \sigma^2_{u0} \end{bmatrix}$
var(use$_{ij}$|π_{ij}) = π_{ij}(1 - π_{ij})/denom$_{ij}$
(2867 of 2867 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The model displayed has the same form as (9.4), but with additional explanatory variables to allow for the effects of **lc**. A new line has appeared, stating that the random effects u_{0j} follow a Normal distribution with mean zero and covariance matrix Ω_u, which for a random intercept model consists of a single term σ^2_{u0}.

- Click on **Start** to fit this model
- When the model has been fitted, click on **Estimates** twice to see the results

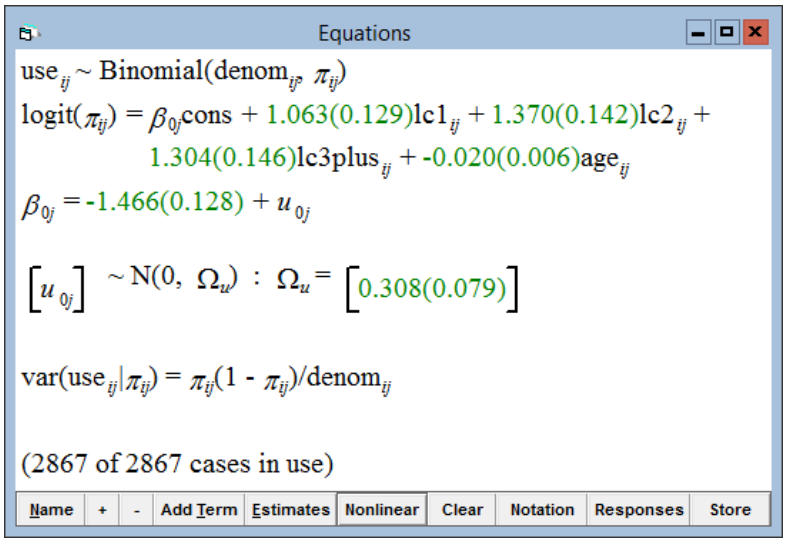

Equations
use$_{ij}$ ~ Binomial(denom$_{ij}$, π_{ij})
logit(π_{ij}) = β_{0j}cons + 0.990(0.126)lc1$_{ij}$ + 1.275(0.138)lc2$_{ij}$ +
1.216(0.142)lc3plus$_{ij}$ + -0.019(0.006)age$_{ij}$
β_{0j} = -1.367(0.123) + u_{0j}
$\begin{bmatrix} u_{0j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} 0.274(0.071) \end{bmatrix}$
var(use$_{ij}$|π_{ij}) = π_{ij}(1 - π_{ij})/denom$_{ij}$
(2867 of 2867 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The above results are obtained using the default estimation procedure, 1st order MQL. As this procedure may lead to estimates that are biased downwards, the 2nd order PQL procedure is preferred. To change the estimation procedure:

- Click on the **Nonlinear** button at the bottom of the **Equations** window
- Under **Linearisation**, select **2nd Order**
- Under **Estimation type**, select **PQL**
- Click **Done**
- Click **More** to fit the model

Note that clicking ***More*** *rather than* ***Start*** *means that the 1st order MQL estimates will be used as starting values in the 2nd order PQL procedure. Because convergence problems may be encountered when using PQL it is advisable to use MQL first and then extend to PQL.*

You should obtain the following estimates that, in this case, are not very different from the 1st order MQL estimates:


Equations
use_ij ~ Binomial(denom_ij, π_ij)
logit(π_ij) = β_0j cons + 1.063(0.129)lc1_ij + 1.370(0.142)lc2_ij +
1.304(0.146)lc3plus_ij + -0.020(0.006)age_ij
β_0j = -1.466(0.128) + u_0j
[u_0j] ~ N(0, Ω_u) : Ω_u = [0.308(0.079)]
var(use_ij|π_ij) = π_ij(1 - π_ij)/denom_ij
(2867 of 2867 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The conclusions regarding the effects of age and number of living children are unchanged by allowing for district-level variation, although the standard errors for the coefficients of the **lc** dummy variables have increased slightly.

The intercept for district j is $-1.466 + u_{0j}$, where the variance of u_{0j} is estimated as 0.308 (SE = 0.079).

For continuous response models, we described how a likelihood ratio test could be used to test the significance of σ^2_{u0}. For discrete response models, estimated using quasi-likelihood methods, the likelihood value is unreliable and so the likelihood ratio test is unavailable. An alternative is to carry out a Wald test, although this test is approximate, as variance parameters are not Normally distributed. A preferred approach is to construct interval estimates for variance parameters using bootstrap or MCMC methods. See

Chapter 3 in Goldstein (2003) and Chapter 4 in Browne (2003). To carry out a Wald test in MLwiN:

- From the **Model** menu, select **Intervals and tests**
- Check **random** at the bottom of the **Intervals and tests** window
- Type a **1** next to **district:cons/cons** (this refers to the parameter σ^2_{u0})
- Click on **Calc**

The test statistic is 15.267, which we compare to a chi-squared distribution on 1 d.f. We therefore conclude that there are significant differences between districts.

Variance partition coefficient

In Chapter 2, we met the variance partition coefficient (VPC) which for a two-level random intercept model is the proportion of total residual variance which is attributable to level 2, i.e, $\sigma^2_{u0}/(\sigma^2_{u0} + \sigma^2_e)$. For a random intercept model fitted to continuous data, the VPC is equal to the *intra-unit correlation*, which is the correlation between two level 1 units in the same level 2 unit. For random coefficient models, the VPC and intra-unit correlation are not equivalent. In the case of binary and other discrete response models, there is no single VPC measure since the level 1 variance is a function of the mean, which depends on the values of the explanatory variables in the model. For example, if y_{ij} is binary then $Var(y_{ij}) = \pi_{ij}(1 - \pi_{ij})$. Therefore the VPC itself will depend on the explanatory variables. Goldstein et al. (2002) propose several alternative approaches for computing a VPC for discrete response data. For those who are interested, a simulation method is described below.

Step 1: From the fitted model, simulate M values for u_{0j} from $\mathrm{N}(0,\hat{\sigma}^2_{u0})$. Denote these simulated values by $u^{(m)}_{0j}$ (m=1, 2, ..., M)

Step 2: For a given value of x_{ij}, x^* say, compute $\pi^{*(m)}_j = \frac{\exp(\hat{\beta}_0+\hat{\beta}_1 x^*+u^{(m)}_{0j})}{1+\exp(\hat{\beta}_0+\hat{\beta}_1 x^*+u^{(m)}_{0j})}$. Also, compute $v^{*(m)}_{1j} = \pi^{*(m)}_j(1 - \pi^{*(m)}_j)$

Step 3: The level 1 variance is then calculated as the mean of the $v^{*(m)}_{1j}$ (m=1, 2, ..., M), and the level 2 variance is the variance of the $\pi^{*(m)}_j$

An MLwiN *macro* (contained in the file **vpc.txt**) has been written to implement this simulation method, with $M = 5000$. We will consider an example shortly.

An alternative VPC measure is obtained if the logistic model is cast in the form of a linear threshold model. We assume that there is a continuous

unobserved variable y^*_{ij} underlying our binary response y_{ij}, such that $y_{ij} = 1$ if $y^*_{ij} \geq 0$ and $y_{ij} = 0$ if $y^*_{ij} < 0$. The unobserved variable y^*_{ij} can be thought of as the propensity to be in one category of the binary response rather than the other, e.g., the propensity to use contraception. The model in (9.4) can be written in terms of y^*_{ij} as

$$y^*_{ij} = \beta_0 + \beta_1 x_{ij} + u_{0j} + e_{ij}$$

where e_{ij} follows a logistic distribution with variance $\pi^2/3 \approx 3.29$. (If a probit link is used, then e_{ij} follows a Normal distribution with variance 1.) So the VPC can be computed as $\sigma^2_{u0}/(\sigma^2_{u0} + 3.29)$. See Snijders & Bosker (1999) for further details.

Note that the two methods of calculating the VPC described above will give different results. This is because the estimate obtained using the simulation method is on the probability scale and depends on covariates while the measure derived from the threshold model is on the logistic scale and hence does not depend on covariates.

We now consider an example. Before running the macro in **vpc.txt**, we have to do the following:

1. set values for the explanatory variables and store these in **c151**, and
2. set values for the explanatory variables which have random coefficients at level 2 and store these in **c152**. (This will be a subset of c151.)

In the model above, we have five explanatory variables (including **cons**). We will begin by computing the VPC for a woman of mean age with no children (i.e, **cons** = 1, **lc1** = 0, **lc2** = 0, **lc3plus** = 0 and **age** = 0). To do this we begin by entering the values (1,0,0,0,0) in **c151**. In a random intercept model, only **cons** has a random coefficient, so we input the value 1 in **c152**. To create these two columns:

- From the **Data Manipulation** menu, select **View or edit data**
- Click on **View**, select **c151**, and click **OK**
- Input the values **1**,**0**,**0**,**0**, and **0** respectively into the first five rows of **c151**
- Click on **View** again, select **c152**, and click **OK**
- Input the value **1** in the first row of **c152**

The macro contains the following sequence of MLwiN commands:

```
► calc c153=(~c151)*.c1098
► pick 1 c153 b2
► calc c153=(~c152)*.omega(2)*.c152
► pick 1 c153 b4
► seed 1
► nran 5000 c154
► calc c154=alog(c154*b4^0.5+b2)
► aver c154 b1 b3 b2
► calc c154=c154*(1-c154)
► aver c154 b5 b1
► calc b8=b2^2/(b1+b2^2)
```

To run this macro:

- From the **File** menu, select **Open Macro**
- Open the file **vpc.txt**
- In the window that shows you these commands, click on **Execute**

The result of running the macro, i.e, the value of the VPC, will be stored in a worksheet box called B8. To print the contents of the box:

- From the **Data Manipulation** menu, select **Command Interface**
- In the space at the bottom of the **Command interface** window, type

```
► print b8
```

- Press **Enter**

You should get a value of approximately 0.048. Therefore, among women of mean age with no children 4.8% of the residual variation is attributable to differences between districts.

To get an idea of the range of the VPC for different values of the explanatory variables, we could compute the VPC for extreme combinations of values. For example, young women with three or more children have a high probability of using contraception, while older women with no children have a low probability of using contraception. The table below gives values for the VPC for these two extreme combinations:

	cons	**lc1**	**lc2**	**lc3plus**	**age** (centred)	**VPC**
High probability of use	1	0	0	1	-9.7	0.069
Low probability of use	1	0	0	0	15.3	0.040

Using a threshold representation of the model, we obtain a VPC of 0.308/(0.308+ 3.290) = 0.086. So approximately 5% to 10% of the residual variance in contraceptive use is attributable to differences between districts.

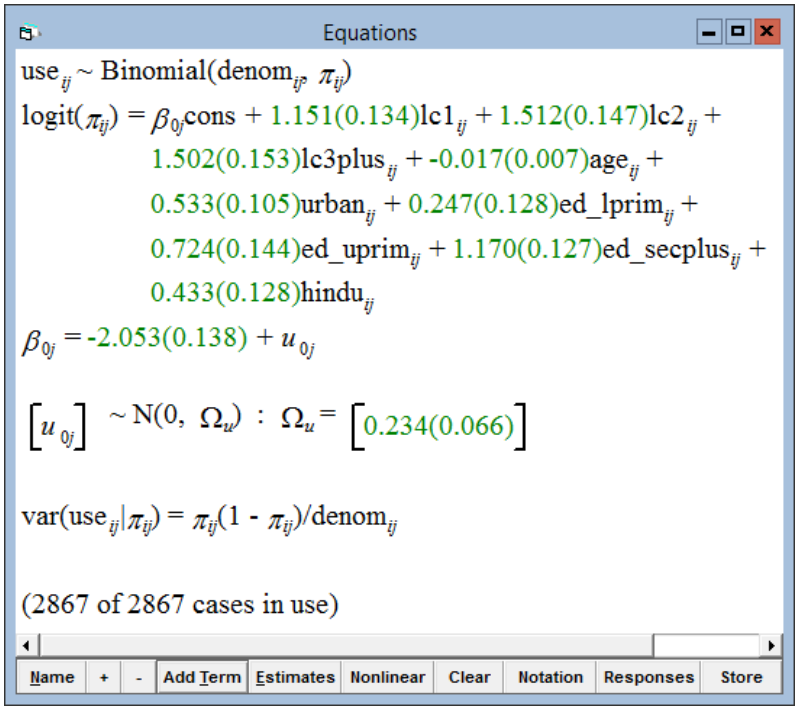
Adding further explanatory variables

We will now add in the remaining woman-level explanatory variables: **urban**, **educ** and **hindu**.

- Use the **Add term** button (and **Specify term** window) three times to add these variables to the model

Education has already been declared as a categorical variable, so dummy variables for three of the categories will be added. Accept the default in which the first category ('no education') is taken as the reference.

When you have fitted this new model (using **More**), you should obtain the following results:


Equations
use_ij ~ Binomial(denom_ij, π_ij)
logit(π_ij) = β_0j cons + 1.151(0.134)lc1_ij + 1.512(0.147)lc2_ij +
1.502(0.153)lc3plus_ij + -0.017(0.007)age_ij +
0.533(0.105)urban_ij + 0.247(0.128)ed_lprim_ij +
0.724(0.144)ed_uprim_ij + 1.170(0.127)ed_secplus_ij +
0.433(0.128)hindu_ij
β_0j = -2.053(0.138) + u_0j
[u_0j] ~ N(0, Ω_u) : Ω_u = [0.234(0.066)]
var(use_ij|π_ij) = π_ij(1 - π_ij)/denom_ij
(2867 of 2867 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The effects of age and number of living children change slightly, but the general conclusions are the same. Higher education levels, living in an urban

area rather than a rural area, and being Hindu rather than Muslim are all positively associated with use of contraception.

Note that ***urban*** *is an individual level variable (as can be seen from the ij subscript), since there are urban and rural areas within a district. Comparing estimated coefficients with their standard errors, we find that all effects are significant at the 5% level.*

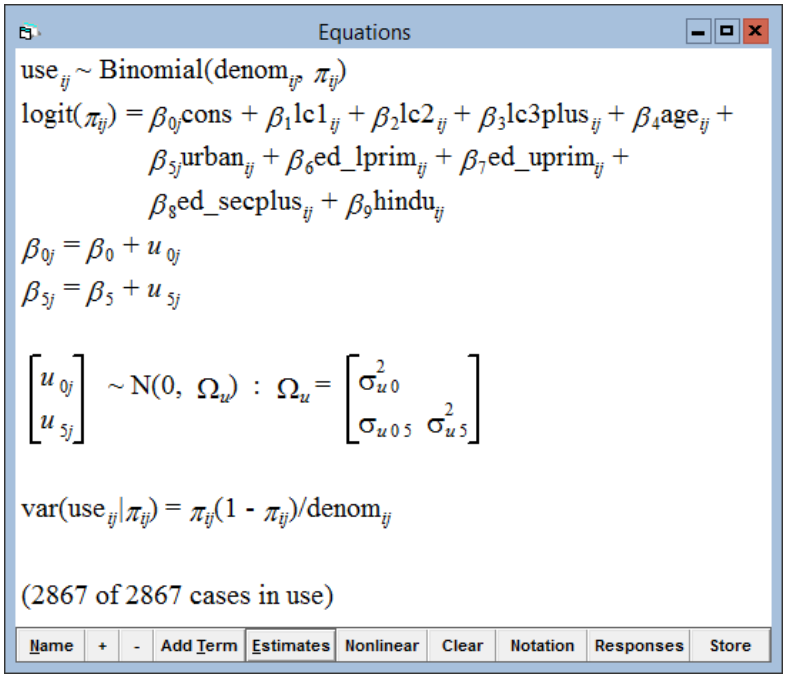
The between-district variance has decreased from 0.308 to 0.234, so some of the variation in contraceptive use between districts is explained by differences in their education, urban/rural and religious composition.

9.4 A two-level random coefficient model

So far, we have allowed the probability of contraceptive use to vary across districts, but we have assumed that the effects of the explanatory variables are the same for each district. We will now modify this assumption by allowing the difference between urban and rural areas within a district to vary across districts. To allow for this effect, we will need to introduce a random coefficient for **urban**. The model we will be fitting has the same form as the random slopes model considered in Chapter 4, but since the variable **urban** has only two categories, we use the more general term *coefficient* rather than *slope* to describe its effect. To introduce a random coefficient for **urban**:

- In the **Equations** window, click on **urban** (or its coefficient)
- In the **X variable** window, check **j(district)** and click **Done**

The model has the following form (you may have to click on **Estimates** to see the notation):


Equations
use$_{ij}$ ~ Binomial(denom$_{ij}$, π_{ij})
logit(π_{ij}) = β_{0j}cons + $\beta_1$lc1$_{ij}$ + $\beta_2$lc2$_{ij}$ + $\beta_3$lc3plus$_{ij}$ + $\beta_4$age$_{ij}$ + β_{5j}urban$_{ij}$ + $\beta_6$ed_lprim$_{ij}$ + $\beta_7$ed_uprim$_{ij}$ + $\beta_8$ed_secplus$_{ij}$ + $\beta_9$hindu$_{ij}$
$\beta_{0j} = \beta_0 + u_{0j}$
$\beta_{5j} = \beta_5 + u_{5j}$
$\begin{bmatrix} u_{0j} \\ u_{5j} \end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u05} & \sigma^2_{u5} \end{bmatrix}$
var(use$_{ij}$|π_{ij}) = π_{ij}(1 - π_{ij})/denom$_{ij}$
(2867 of 2867 cases in use)
Name
+
-
Add Term
Estimates
Nonlinear
Clear
Notation
Responses
Store

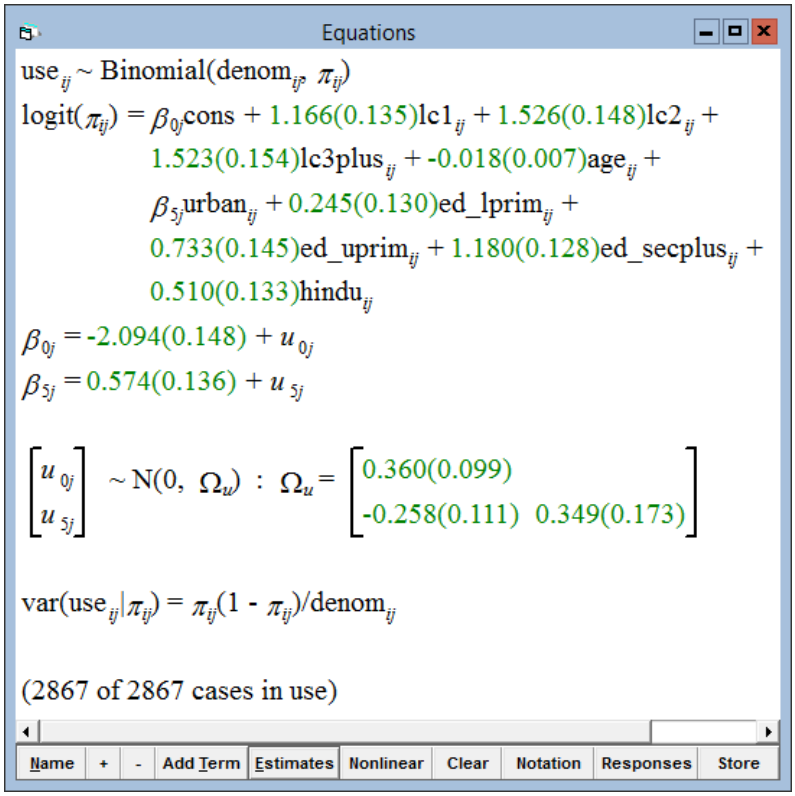
A j subscript has been added to the coefficient of **urban**, indicating that the coefficient depends on district. The average effect of **urban** is β_5, but the effect for district j is $\beta_{5j} = \beta_5 + u_{5j}$ where u_{5j} is a Normally distributed random effect with mean zero and variance σ^2_{u5}. Allowing the coefficient of **urban** to vary across districts has also introduced the parameter σ_{u50}, which is the covariance between u_{0j} and u_{5j}.

As for continuous response random coefficient models, the level 2 variance is a function of the explanatory variables that have random coefficients. In Chapter 7, we met variance functions and the same ideas can be applied to any multilevel model. For the model specified above, the residual variance between districts is a function of **urban**:

$$\begin{aligned} \text{var}(u_{0j} + u_{5j}\mathbf{urban}) &= \text{var}(u_{0j}) + 2\text{cov}(u_{0j}, u_{5j})\mathbf{urban} + \text{var}(u_{5j})\mathbf{urban}^2 \\ &= \sigma^2_{u0} + (2\sigma_{u50} + \sigma^2_{u5})\mathbf{urban} \end{aligned} \tag{9.5}$$

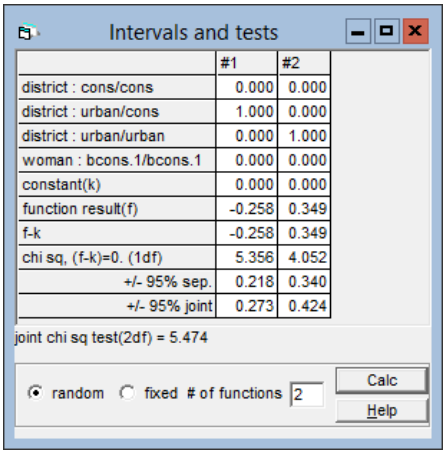
Note that because **urban** *is a (0,1) variable,* $\mathbf{urban}^2 = \mathbf{urban}$*. For rural areas (*$\mathbf{urban} = 0$*), the residual district level variance is* σ^2_{u0}*. For urban areas (*$\mathbf{urban} = 1$*), the residual district level variance is* $\sigma^2_{u0} + 2\sigma_{u50} + \sigma^2_{u5}$*.*

- Click **More** to fit the random coefficient model
- Click **Estimates** to see the estimated coefficients and their standard errors


Equations
use$_{ij}$ ~ Binomial(denom$_{ij}$, π_{ij})
logit(π_{ij}) = β_{0j}cons + 1.166(0.135)lc1$_{ij}$ + 1.526(0.148)lc2$_{ij}$ +
1.523(0.154)lc3plus$_{ij}$ + -0.018(0.007)age$_{ij}$ +
β_{5j}urban$_{ij}$ + 0.245(0.130)ed_lprim$_{ij}$ +
0.733(0.145)ed_uprim$_{ij}$ + 1.180(0.128)ed_secplus$_{ij}$ +
0.510(0.133)hindu$_{ij}$
β_{0j} = -2.094(0.148) + u_{0j}
β_{5j} = 0.574(0.136) + u_{5j}
[u_{0j} u_{5j}] ~ N(0, Ω_u) : Ω_u = [0.360(0.099) / -0.258(0.111) 0.349(0.173)]
var(use$_{ij}$|π_{ij}) = π_{ij}(1 - π_{ij})/denom$_{ij}$
(2867 of 2867 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

We can test the significance of the added parameters, σ^2_{u5} andσ_{u50}, using a Wald test:

- From the **Model** menu, select **Intervals and tests**
- Choose **random** in the **Intervals and tests** window
- Next to **# of functions**, type **2**
- Edit the values next to **district: urban/cons** and **district: urban/urban** as shown below
- Click **Calc**



	#1	#2
district : cons/cons	0.000	0.000
district : urban/cons	1.000	0.000
district : urban/urban	0.000	1.000
woman : bcons.1/bcons.1	0.000	0.000
constant(k)	0.000	0.000
function result(f)	-0.258	0.349
f-k	-0.258	0.349
chi sq, (f-k)=0. (1df)	5.356	4.052
+/- 95% sep.	0.218	0.340
+/- 95% joint	0.273	0.424

The test statistic is 5.471, which is approximately chi-squared distributed on 2 d.f. ($p = 0.065$). At the 10% level, we conclude that both parameters

are non-zero, which implies that the effect of **urban** does indeed vary across districts.

On average (after adjusting for the effects of age and the other explanatory variables), the log-odds of using contraception are 0.574 higher for urban areas than for rural areas. Depending on the value of u_{5j}, the difference in a given district will be larger or smaller than 0.574.

Substituting the estimates of σ^2_{u0}, σ^2_{u5} and σ_{u50} into (9.5), we obtain the following estimates of residual district-level variation:

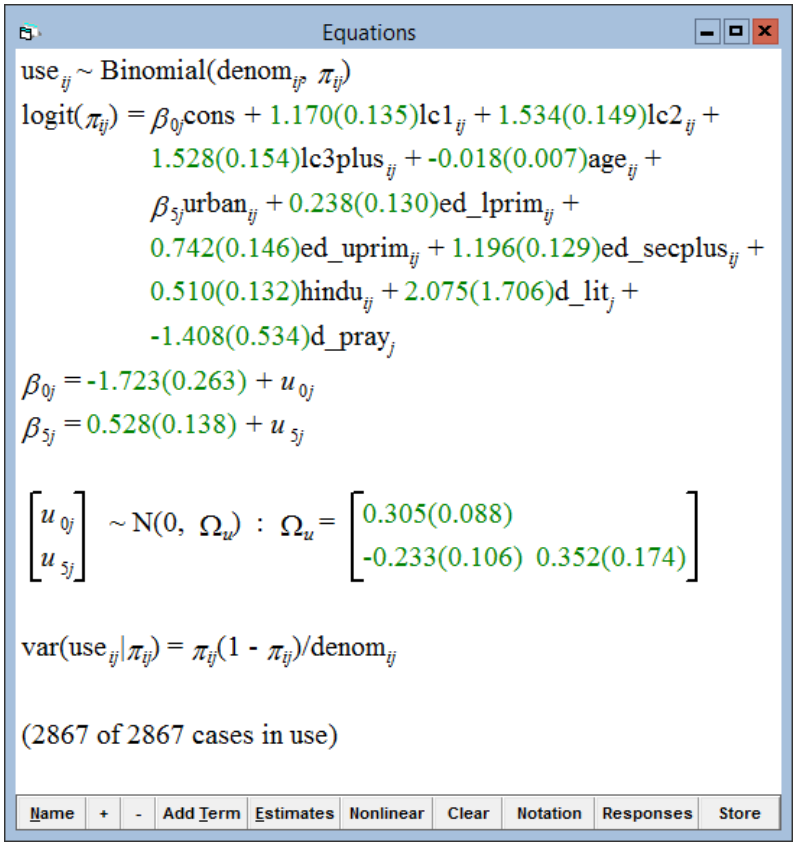
For rural areas: district-level variance = 0.360

For urban areas: district-level variance $= 0.360 + 2(-0.258) + 0.349 = 0.193$

So there is greater district-level variation in the probability of using contraception in rural areas than in urban areas.

We will now add in our two district-level explanatory variables, **d_lit** and **d_pray**, to see whether they explain some of the district-level variation in urban and rural areas. Use the **Add term** button to add both variables to the model, and click **More** to fit the model.

You should get the following results:


Equations
use_ij ~ Binomial(denom_ij, π_ij)
logit(π_ij) = β_0j cons + 1.170(0.135)lc1_ij + 1.534(0.149)lc2_ij +
1.528(0.154)lc3plus_ij + -0.018(0.007)age_ij +
β_5j urban_ij + 0.238(0.130)ed_lprim_ij +
0.742(0.146)ed_uprim_ij + 1.196(0.129)ed_secplus_ij +
0.510(0.132)hindu_ij + 2.075(1.706)d_lit_j +
-1.408(0.534)d_pray_j
β_0j = -1.723(0.263) + u_0j
β_5j = 0.528(0.138) + u_5j
[u_0j; u_5j] ~ N(0, Ω_u) : Ω_u = [0.305(0.088); -0.233(0.106) 0.352(0.174)]
var(use_ij|π_ij) = π_ij(1 - π_ij)/denom_ij
(2867 of 2867 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The effect of the proportion of literate women in the district has a positive, but non-significant effect on the probability of using contraception. District-level religiosity has a significant effect, with women living in districts with higher levels of religiosity being less likely to use contraception.

The residual between-district variation is now 0.305 for rural areas and $0.305 + 2(-0.233) + 0.352 = 0.191$ for urban areas. Some of district-level variation in rural areas is explained by differences in religiosity, but the variation in urban areas is almost unchanged.

9.5 Modelling binomial data

So far, we have considered logistic models for binary response data, but the same models may be used to analyse binomial data, where the response variable is a proportion. For illustration, we will convert the binary woman-level contraceptive use variable to district-level proportion data. We will then model the proportion of contraceptive users in a district as a function of the district-level explanatory variables and district-level random effects. Of course in practice, since in this case we have individual-level data and we know that there are important individual-level predictors of contraceptive use, we would not want to aggregate the data in this way. If, however, we only had access to aggregate data then it is more efficient to model the proportions directly rather than converting to individual-level binary responses.

Modelling district-level variation with district-level proportions

Our response variable y_j will be the sample proportion of contraceptive users in district j. After aggregating our data to the district level, the only other change to the model is that the denominator n_j will no longer equal 1 as for binary data, but will equal the number of women of reproductive age in district j.

Although our response variable is now at the district level, we can still fit a two-level random intercept model of the form:

$$\begin{aligned} \text{logit}(\pi_{ij}) &= \beta_{0j} + \beta_1 \mathbf{d\_lit}_j + \beta_2 \mathbf{d\_pray}_j \\ \beta_{0j} &= \beta_0 + u_{0j} \end{aligned}$$

where π_{ij} is the probability of using contraception for woman i in district j as before. When we specify the model, we will use the aggregate district

ID as the identifier for both level 1 and level 2. This implies a model with 60 level 2 units (districts), each with one level 1 observation. This might appear, at first glance, to produce a confounded model. However, we should remember that each level 1 unit has an associated denominator n_j, which is the number of women in the district. It is this associated woman-level information, together with the fact that the level 1 variance depends on the explanatory variables in the model, which prevents the model from being confounded [3].

Creating a district-level data set

First we need to clear the current model settings.

- In the **Equations** window, click on the **Clear** button
- From the **Data Manipulation** menu, select **Command interface**
- At the bottom of the **Command interface** window, type:

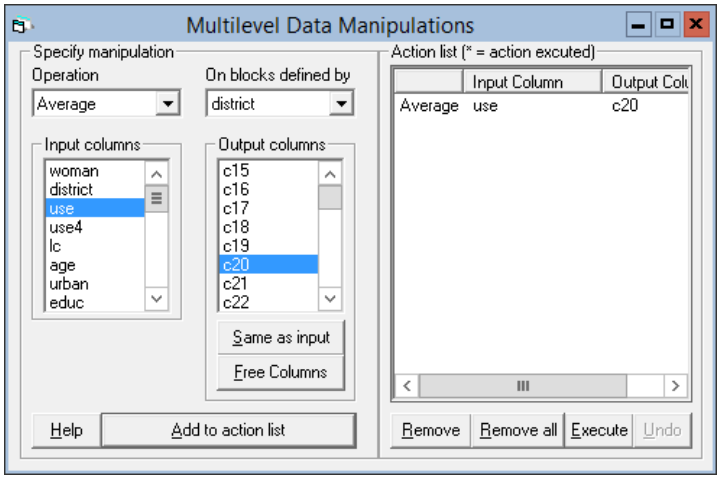
```
► erase 'denom' 'cons'
```

- Press **return**

We will now create a district-level data set. We will begin by creating the response variable y_j and the denominator n_j:

- From the **Data Manipulation** menu, select **Multilevel data manipulations**
- In the **Multilevel Data Manipulations** window, under **Operation**, select **Average**
- For **On blocks defined by**, select **district**
- For **Input columns**, select **use**
- For **Output columns**, select **c20**
- Click **Add to action list**.
- Check that your window looks like the one below, then click **Execute**
- Then change **Operation** to **Count** (Don't worry about what happens in the **Input columns** section)
- For **Output columns**, select **c21**
- Click **Add to action list**, followed by **Execute**

[3] If the n_j were equal across districts and no explanatory variables were included in the model, then it would not be possible to identify district-level variation.

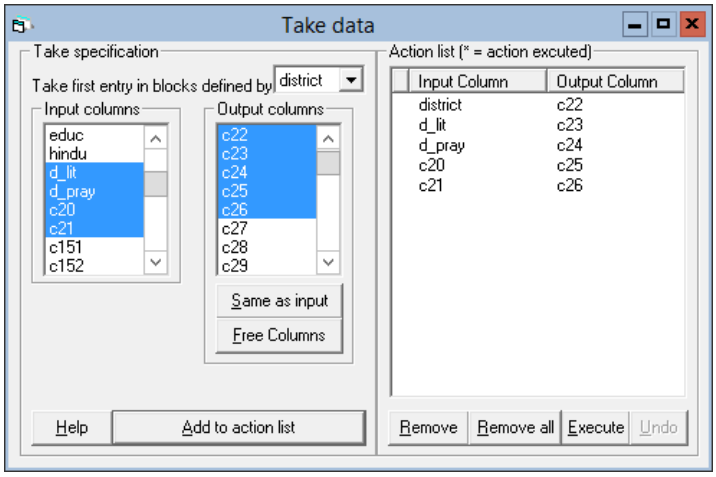


The columns **c20** and **c21** contain the response variable and denominator respectively. However, they still contain a record for each woman, where the values for women in the same district are replicated. To see this

- From the **Data Manipulation** menu, select **View or edit data**
- Select **district**, **c20** and **c21** (Use ctrl-click to make multiple selections.)

We will next convert **c20** and **c21** so that they have one record per district, and at the same time create district-level versions of **district**, **d_lit** and **d_pray**.

- From the **Data Manipulation** menu, select **unreplicate** to open the **Take data** window
- For **Take first entry in blocks defined by**, select **district** from the drop-down list
- For **Input columns**, select the variables **district**, **d_lit**, **d_pray**, **C20** and **C21** (using ctrl-click)
- For **Output columns**, select **c22-c26**
- Click **Add to action list**.
- Check that your window looks like the one below, then click on **Execute**



- Using the **Names** window, give the following names to **c22-c26** (in order): **district1**, **d_lit1**, **d_pray1**, **prop** and **denom**

The final step in setting up the data is to create a new **cons** variable:

- From the **Data Manipulation** menu, select **Generate vector**
- In the Generate Vector window, check **Constant vector**
- For the **Output column**, select **C27**
- For **Number of copies**, type **60** (the number of districts)
- For **Value**, type **1**
- Click **Generate**
- In the **Names** window, give **c27** the name **cons**

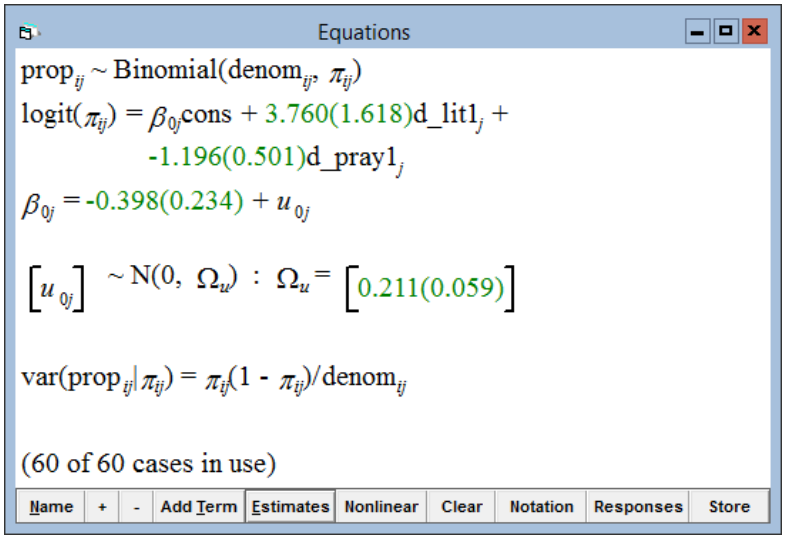
Fitting the model

We can now set up the model:

- In the **Equations** window, first click on y. In the **Y variable** window, select:
  y: **prop**
  N levels: **2-ij**
  Level 2(j): **district1**
  Level 1(i): **district1**
- Click **done**, and return to the **Equations** window
- Click on N. In the **Response type** window, select **binomial** and **logit**, then Click **Done**

- Click on n_{ij}, and in the **specify denominator** window, select **denom**. Click **Done**
- Click on x_0
- Select **cons** from the **X variable** window's drop-down list, and check both **Fixed parameter** and **j(district1)**
- Click **Done**
- Click **Add term** and select **d_lit1**. Click **Done**
- Click **Add term** and select **d_pray1**. Click **Done**
- Click **Nonlinear** and check **Use Defaults** (1st order MQL). Click **Done**
- Click **Start** to fit the model

Now change to 2nd order PQL using the **Nonlinear** button, and Click **More**. You should get the following results:


Equations
prop_ij ~ Binomial(denom_ij, π_ij)
logit(π_ij) = β_0j cons + 3.760(1.618)d_lit1_j +
-1.196(0.501)d_pray1_j
β_0j = -0.398(0.234) + u_0j
[u_0j] ~ N(0, Ω_u) : Ω_u = [0.211(0.059)]
var(prop_ij|π_ij) = π_ij(1 - π_ij)/denom_ij
(60 of 60 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

Note that we would have got exactly the same results had we fitted a random intercepts model to the binary response variable ***use****, with only an intercept plus the district level explanatory variables* ***d_lit*** *and* ***d_pray****.*

Chapter learning outcomes

- ⋆ How to specify a binary response model in MLwiN via the Equations window

- ⋆ The fact that standard likelihood methods cannot be used for binary response models, so quasi-likelihood methods are used instead
- ⋆ The interpretation of fixed effects is more complicated in binary response models
- ⋆ How to fit a general Binomial model where the response is a proportion

Chapter 10

Multinomial Logistic Models for Unordered Categorical Responses

10.1 Introduction

In many studies, the response variable of interest is categorical. In Chapter 9 we considered multilevel models for binary categorical responses. The logistic models described there may be extended to permit response variables with more than two categories, but the type of model we fit depends on whether the categories are ordered or unordered. Examples of ordered responses are attitude scales (e.g. with categories going from 'strongly disagree' to 'strongly agree') and exam grades. Examples of unordered responses include political affiliation and cause of death. In this chapter, we introduce the multinomial logistic model for unordered categorical responses. In Chapter 11 we examine models for ordered responses.

We will now re-analyse the contraceptive use data set from Bangladesh (in **bang.ws**) that was introduced in Chapter 9. In our earlier analysis, the response variable was a binary indicator for use of contraception at the time of the survey. In any serious study of contraceptive behaviour, however, we would wish to distinguish between different methods of contraception, particularly between modern or efficient methods (e.g. pills and IUDs) and traditional or inefficient methods (e.g. withdrawal). In this chapter, our response is an unordered categorical variable that distinguishes between different types of method among users. Our response variable is **use4**, which is coded as follows:

use4	Contraceptive use status and method
1	Sterilization (male or female)
2	Modern reversible method
2	Traditional method
4	Not using contraception

All other variables in **bang.ws** are described in Section 9.1.

To see the frequency distribution of **use4**, open **bang.ws** in MLwiN, then

- From the **Basic Statistics** menu, select **Tabulate**
- Check **Percentages of column totals**
- Next to **Columns**, select **use4** from the drop-down list
- Click **Tabulate**

You will see that 10.5% of women (or their husbands) were sterilized, 19.4% were using a modern reversible method (mainly pills in Bangladesh), 9.8% were using a traditional method, and 60.3% were not using any contraception. In our analysis, we will be interested in determining the factors associated with use of these different types of method (or non-use).

10.2 Single-level multinomial logistic regression

Suppose that y_i is the unordered categorical response for individual i, and that the response variable has t categories. We denote the probability of being in category s by $\pi_i^{(s)} = \Pr(y_i = s)$. In a multinomial logistic model, one of the response categories is taken as the reference category, just as the category coded '0' is usually taken as the reference category in a binary response model. A set of t-1 equations is then estimated, contrasting each of the remaining response categories with the chosen reference category. Suppose that the last category is taken as the reference. Then, for a single explanatory variable x_i, a multinomial logistic regression model with logit link is written:

$$\log\left(\frac{\pi_i^{(s)}}{\pi_i^{(t)}}\right) = \beta_0^{(s)} + \beta_1^{(s)} x_i, \qquad s = 1, \ldots, t-1 \tag{10.1}$$

A separate intercept and slope parameter is usually estimated for each contrast, as indicated by the s superscripts, although it is possible to constrain some or all to be equal. In the model above, the same explanatory variable appears in each of the $t-1$ contrasts. Although this is usual practice, and a

requirement in some software packages, models where the set of explanatory variables differs across contrasts can be estimated in MLwiN.

The parameter $\beta_1^{(s)}$ is interpreted as the additive effect of a 1-unit increase in x on the log-odds of being in category s rather than category t. As in the binary logit model, it is more meaningful to interpret $\exp(\beta_1^{(s)})$, which is the multiplicative effect of a 1-unit increase in x on the odds of being in category s rather than category t. However, an easier way to interpret the effect of x is to calculate predicted probabilities $\pi_i^{(s)}(s = 1, \ldots, t)$ for different values of x.

The following expression for $\pi_i^{(s)}$ $(s = 1, \ldots, t-1)$ can be derived from (10.1):

$$\pi_i^{(s)} = \frac{\exp(\beta_0^{(s)} + \beta_1^{(s)} x_i)}{1 + \sum_{k=1}^{t-1} \exp(\beta_0^{(k)} + \beta_1^{(k)} x_i)} \tag{10.2}$$

The probability of being in the reference category t is obtained by subtraction:

$$\pi_i^{(t)} = 1 - \sum_{k=1}^{t-1} \pi_i^{(k)} \tag{10.3}$$

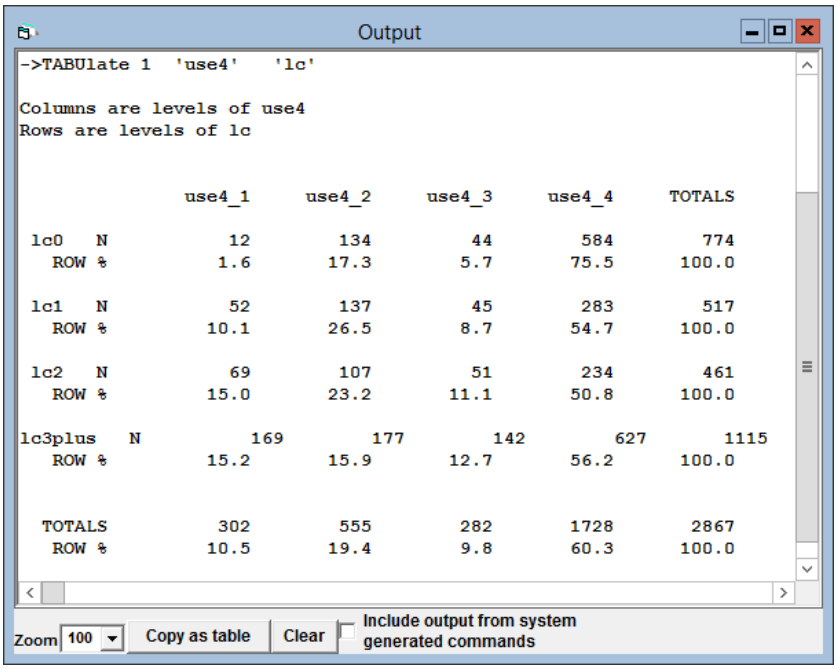
Model interpretation using both odds ratios and predicted probabilities will be considered in the example that follows.

10.3 Fitting a single-level multinomial logistic model in MLwiN

We will begin by fitting a model with a single covariate, the number of living children (**lc**). First, we will look at a cross-tabulation of **use4** and **lc** to see how the decision to use contraception and the choice of method depends on number of children in our sample.

- From the **Basic Statistics** menu, select **Tabulate**
- Check **Percentages of row totals**
- Next to **Columns**, select **use4** from the drop-down list
- Check **Rows**, and select **lc** from the drop-down list
- Click **Tabulate**

You should see this table in the **Output** window:

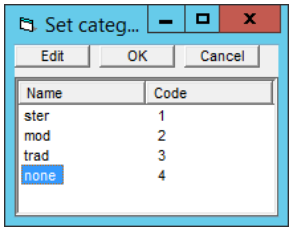

Output
->TABUlate 1 'use4' 'lc'
Columns are levels of use4
Rows are levels of lc
use4_1 use4_2 use4_3 use4_4 TOTALS
lc0 N 12 134 44 584 774
ROW % 1.6 17.3 5.7 75.5 100.0
lc1 N 52 137 45 283 517
ROW % 10.1 26.5 8.7 54.7 100.0
lc2 N 69 107 51 234 461
ROW % 15.0 23.2 11.1 50.8 100.0
lc3plus N 169 177 142 627 1115
ROW % 15.2 15.9 12.7 56.2 100.0
TOTALS 302 555 282 1728 2867
ROW % 10.5 19.4 9.8 60.3 100.0
Zoom 100
Copy as table
Clear
Include output from system generated commands

In Chapter 9, we saw that the probability of contraceptive use was much higher among women with one or more child than among those without children. Here we see that, among contraceptive users, the type of method chosen also varies with number of children. For example, as would be expected, women with no children are unlikely to choose sterilization. Women with one or two children are the most likely to use a modern reversible method; the lower probability of modern reversible use among women with three or more children is likely to be due to factors associated with high fertility.

We will now model this relationship using a multinomial logistic regression model of the same form as (10.1). As in Chapter 9, we will include as explanatory variables three dummy variables for **lc**, taking the first category 'no children' as the reference.

We will start by declaring **use4** to be categorical and attaching labels to its categories:

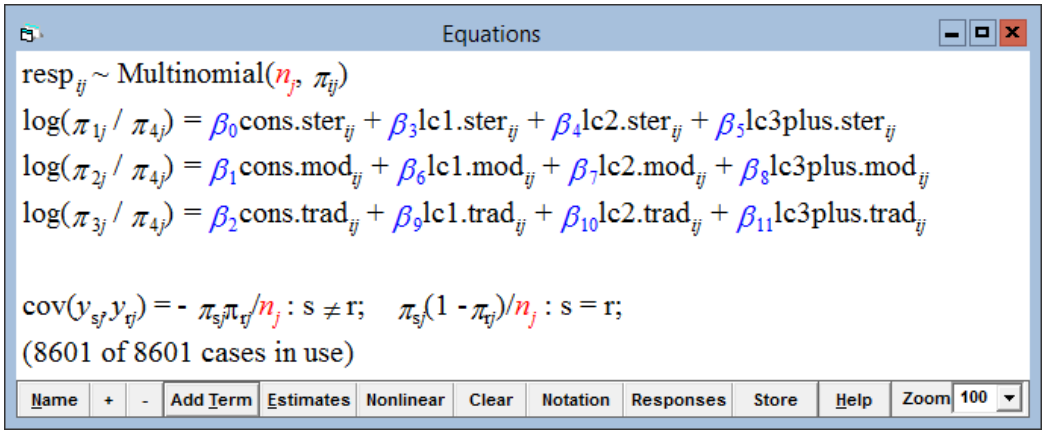
- From the **Data Manipulation** menu, select **Names**
- In the **Names** window, highlight **use4** and click on **View** in the categories section
- For category **1**, type **ster** in the **name** column
- Give the following names to categories **2**, **3** and **4**: **mod**, **trad** and **none** (as shown in the figure below)
- Click **OK**



We can now set up the model:

- In the **Equations** window, click on the **Name** button
- Click on y and make the following selections in the **Y-variable** window:
  y: **use4**
  N levels: **1-i**
  level 1(i): **woman**
- Click **done**
- Click on the N in the **Equations** window
- In the **Response type** window, scroll down and select **Multinomial**
- We will use the defaults of **logit** link function and the **unordered** multinomial option
- Next to **ref category:**, select **none** from the drop-down list
- Click **Done**
- Click on **Add term** and, under **variable**, select **cons**
- Click **add Separate coefficients**
- Click on **Add term** again, select **lc**, and click **add Separate coefficients**
- Click **Name**
- Click **Estimates**

The **Equations** window should look like this:


Equations
$resp_{ij} \sim \text{Multinomial}(n_j, \pi_{ij})$
$\log(\pi_{1j} / \pi_{4j}) = \beta_0 \text{cons.ster}_{ij} + \beta_3 \text{lc1.ster}_{ij} + \beta_4 \text{lc2.ster}_{ij} + \beta_5 \text{lc3plus.ster}_{ij}$
$\log(\pi_{2j} / \pi_{4j}) = \beta_1 \text{cons.mod}_{ij} + \beta_6 \text{lc1.mod}_{ij} + \beta_7 \text{lc2.mod}_{ij} + \beta_8 \text{lc3plus.mod}_{ij}$
$\log(\pi_{3j} / \pi_{4j}) = \beta_2 \text{cons.trad}_{ij} + \beta_9 \text{lc1.trad}_{ij} + \beta_{10} \text{lc2.trad}_{ij} + \beta_{11} \text{lc3plus.trad}_{ij}$
$\text{cov}(y_{sj}, y_{rj}) = -\pi_{sj}\pi_{rj}/n_j : s \neq r; \quad \pi_{sj}(1-\pi_{rj})/n_j : s = r;$
(8601 of 8601 cases in use)
Name
+
-
Add Term
Estimates
Nonlinear
Clear
Notation
Responses
Store
Help
Zoom
100

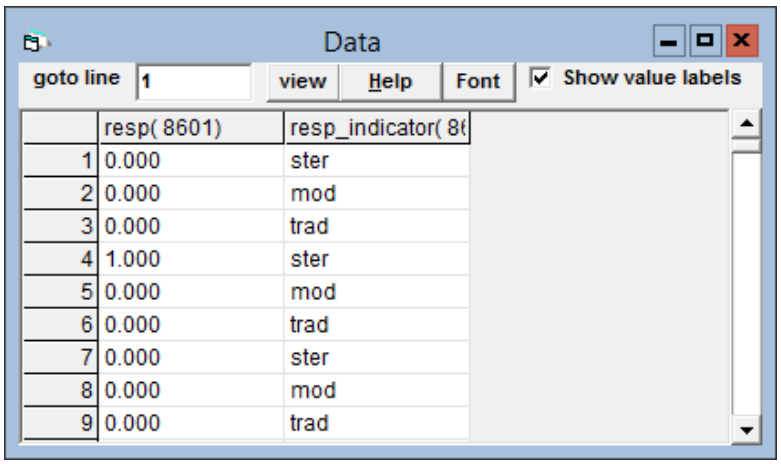
Before going any further, we will take a moment to explain the notation

used in the **Equations** window and how it relates to equation (10.1). The model we have specified above has the same form as (10.1), but with three explanatory variables for the effects of **lc**. Notice, however, that although we have specified a single-level model, all variables have two subscripts, i and j. In MLwiN the single-level multinomial model is framed as a two-level model, with response categories at level 1 and individuals at level 2.

The categorical variable **use4** has been transformed to three binary variables (corresponding to categories 1, 2 and 3). These are stacked to form a new response variable named **resp**, a column that has $3 \times 2867 = 8601$ observations. This new variable has a two-level structure with three binary responses per woman.

If you click on **resp** $_{ij}$ in the **Equations** window, you will see that **N levels:** has changed to **2-ij**, with **woman_long** as the level 2 ID and **resp_indicator** as the level 1 ID. The variable **woman_long** is a 'long' version of **woman**, with each value of **woman** copied three times to obtain a column that is the same length as **resp**. The variable **resp_indicator** is also automatically created when a multinomial model is specified in MLwiN.

To see how **resp** and **resp_indicator** are constructed, look at the values of **use4** for the first three women. They are 4 (none), 1 (ster) and 4 (none). Now look at the first few values of the variables **resp** and **resp_indicator**:



	resp(8601)	resp_indicator(8(
1	0.000	ster
2	0.000	mod
3	0.000	trad
4	1.000	ster
5	0.000	mod
6	0.000	trad
7	0.000	ster
8	0.000	mod
9	0.000	trad

For each woman, we have three values of **resp**, corresponding to categories 1 (ster), 2 (mod) and 3 (trad), respectively. The reference category (none) is omitted. These values are stacked, and a category indicator is stored in **resp_indicator**. The variable **resp** has two subscripts — to index the response category (i) and the woman (j). For woman j, **resp** $_{ij}$= 1 if **use4** = i, and 0 otherwise (i= 1, 2, 3). For example, for a woman using a modern reversible method (category 2) the three values of **resp** are 0, 1 and 0. As we saw in the **Data** window, the first woman in the data set is not using contraception, so her values of **resp** are (0, 0, 0). The second woman is sterilized, so she has values (1, 0, 0).

Returning to the **Equations** window, the first line says that the binary variable **resp** follows a multinomial distribution, which has parameters n_j and π_{ij}. As in Chapter 9, we have a denominator n_j which must be specified. In the case where the individual is our lowest level of observation and we have one observation per individual, $n_j = 1$. More generally, our responses may be proportions, for example the proportion in each category of **use4** in an area. In that case, n_j would be the population size in area j.

Parameter π_{ij} is the probability (predicted by the model, from individual j's pattern of explanatory variables) that individual j is in response category i. The remaining lines in the **Equations** window specify three pairwise contrasts between each of the response categories 1 (ster), 2 (mod) and 3 (trad) and the reference category 4 (none). Each equation includes a constant term, **cons***, and three dummy variables for number of living children, **lc1***, **lc2*** and **lc3plus***, where the suffix (the replacement for *) indicates the response category being contrasted with the reference category in that equation.

To illustrate how these variables are created, we will consider the first contrast (sterilization vs. none). First **cons.ster** is constructed from **resp_indicator** as follows:

$$\begin{aligned} \textbf{cons.ster} &= 1 \text{ if } \textbf{resp\_indicator} = 1 \text{ (i.e, ster)} \\ &= 0 \text{ otherwise} \end{aligned}$$

Then 'long' versions of **lc1**, **lc2** and **lc3plus** are created by repeating each of their values three times, in the same way that **woman_long** was created from **woman**. Each of these 'long' variables is then multiplied by **cons.ster** to obtain **lc1.ster**, **lc2.ster** and **lc3plus.ster**.

Although the same set of variables is included in each contrast, it is possible to exclude an explanatory variable from one or more contrasts. To remove an explanatory variable, click on the variable in the equation for the contrast from which it is to be excluded and click on the **X variable** window's **Delete term** button.

The last line in the **Equations** window shows the terms in the variance-covariance matrix for $\textbf{resp}_{ij}$. See Chapter 4 in Goldstein (2003) for further details.

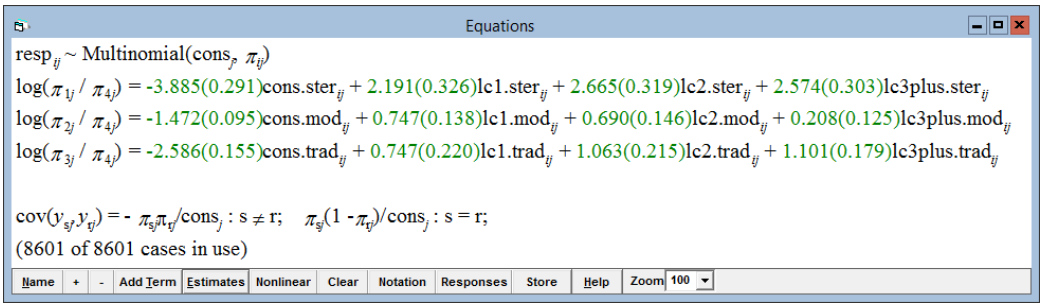
To complete the model specification, we need to declare n_j (see Section 9.2). For a binary response model, n_j is a vector of 1s. The constant vector **cons** has the required structure.

- In the **Equations** window, click on n_j and select **cons** from the drop-down list
- Click **Done**

To fit the model:

- Click on the **Nonlinear** button at the bottom of the **Equations** window
- Click on **Use defaults**, then **Done**
- Click on **Start**

After clicking on **Estimates** twice, you should see the following output:


Equations
resp_ij ~ Multinomial(cons_j, π_ij)
log(π_1j / π_4j) = -3.885(0.291)cons.ster_ij + 2.191(0.326)lc1.ster_ij + 2.665(0.319)lc2.ster_ij + 2.574(0.303)lc3plus.ster_ij
log(π_2j / π_4j) = -1.472(0.095)cons.mod_ij + 0.747(0.138)lc1.mod_ij + 0.690(0.146)lc2.mod_ij + 0.208(0.125)lc3plus.mod_ij
log(π_3j / π_4j) = -2.586(0.155)cons.trad_ij + 0.747(0.220)lc1.trad_ij + 1.063(0.215)lc2.trad_ij + 1.101(0.179)lc3plus.trad_ij
cov(y_sj, y_tj) = - π_sj π_tj/cons_j : s ≠ r; π_sj(1 - π_tj)/cons_j : s = r;
(8601 of 8601 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

We can carry out approximate significance tests on the coefficients of the dummy variables for **lc** by dividing the estimated coefficients by their standard errors and comparing these quotients to a unit Normal distribution. Here, the coefficients of **lc1**, **lc2** and **lc3plus** are all large relative to their standard errors, so we conclude that having children shows a statistically significant effect on the probability of using *any* type of contraceptive method.

To interpret the effects of **lc** on contraceptive choice, we take exponentials of the estimated coefficients of **lc1**, **lc2**, and **lc3plus** to obtain odds ratios as follows:

	Ster. Vs. None		Mod. vs. None		Trad. vs. None	
Category of **lc**	β	$\exp(\beta)$	β	$\exp(\beta)$	β	$\exp(\beta)$
None	0	1	0	1	0	1
1	2.191	8.94	0.747	2.11	0.747	2.11
2	2.665	14.37	0.690	1.99	1.063	2.90
3+	2.574	13.12	0.208	1.23	1.101	3.01

From the odds ratios we can see, for example, that the probability of choosing sterilization increases sharply as **lc** changes from no children to one child. The odds of using sterilization rather than no method are 8.95 times higher for women with one child than for women with no children.

Note that this odds ratio could have been obtained directly from the cross tabulation of **use4** and **lc**. For example:

$$\frac{\text{(\# with 1 child using sterilisation) / (\# with 1 child using no method)}}{\text{(\# with 0 children using sterilisation) / (\# with 0 children using no method)}}$$
$$= \frac{52/283}{12/584} = 8.94$$

For any method, the probability of use increases as the number of children increases from 0 up to 2. There is a slight decrease in the probability of using a modern method (either permanent or reversible) for women with three or more children, compared to women with two children.

When there are several response categories, it is often easier to interpret a fitted model by calculating predicted probabilities for different values of an explanatory variable, while holding constant the values of other explanatory variables. These probabilities are calculated using (10.2) and (10.3). For the simple model above, a macro (in the file **predprob.txt**) has been written to compute predicted probabilities for a given category of **lc**.

To use these macros, we need to input values for the dummy variables **lc1**, **lc2** and **lc3plus** into **c50**. We will begin by computing probabilities of contraceptive use for women with no children (i.e, **lc1** = 0, **lc2** = 0, **lc3plus** = 0).

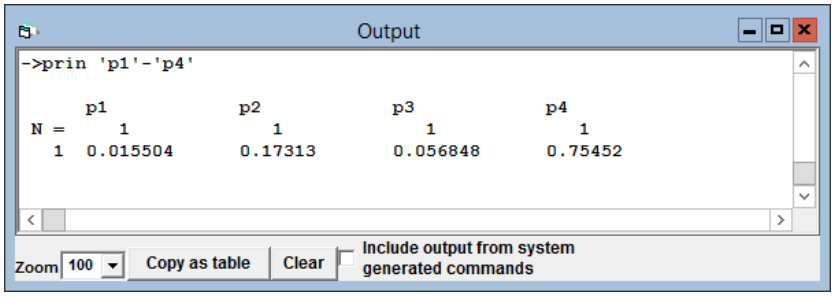
- From the **Data Manipulation** menu, select **View or edit data**
- Click on **View**, select **c50**, and Click **OK**
- Input the values **0**, **0**, and **0** into the first 3 rows of **c50**

To run the macro:

- From the **File** menu, select **Open Macro**
- Open the file **predprob.txt**
- In the window that opens showing the macro commands, click on **Execute**
- The predicted probabilities of being in categories 1 to 4 of **use4** will be stored in columns labeled **p1** to **p4**
- From the **Data Manipulation** menu, select **Command Interface**
- Type

```
► print 'p1'-'p4'
```

You should get the following values:



Notice that these are the same as the sample proportions in each category of **use4** for **lc** = 0 (see the cross-tabulation of **use4** and **lc**). Changing the values in **c50** to (1,0,0) and rerunning the macro will give predicted probabilities corresponding to **lc** = 1. We leave this as an exercise for the reader.

10.4 A two-level random intercept multinomial logistic regression model

The data in **bang.ws** have a two-level hierarchical structure with women at level 1, nested within districts at level 2. In Chapter 9, the single-level model for the binary response **use** was extended to allow for district effects on the probability of using contraception. In a similar way, the single-level model for unordered categorical responses such as **use4** can be extended to two levels.

Suppose that y_{ij} is the categorical response for individual i in district j, and denote the probability of being in category s by $\pi_{ij}^{(s)}$. Equation (10.1) can be extended to a two-level random intercept model:

$$\log\left(\frac{\pi_{ij}^{(s)}}{\pi_{ij}^{(t)}}\right) = \beta_0^{(s)} + \beta_1^{(s)} x_{ij} + u_j^{(s)}, \qquad s = 1, ..., t-1 \tag{10.4}$$

where $u_j^{(s)}$ is a district-level random effect, assumed to be Normally distributed with mean 0 and variance $\sigma_u^{2(s)}$. The random effects are contrast-specific, as indicated by the s superscript, because different unobserved district-level factors may affect each contrast. Or, equivalently, the intra-district correlation in contraceptive use may vary by type of method. However, the random effects may be correlated across contrasts: $\text{cov}(u_j^{(s)}, u_j^{(r)}) = \sigma_u^{(s,r)}, s \neq r$. Correlated random effects would arise, for example, if there were unobserved district-level factors which affect the choice of more than one method type.

As with binary response models, different procedures have been implemented

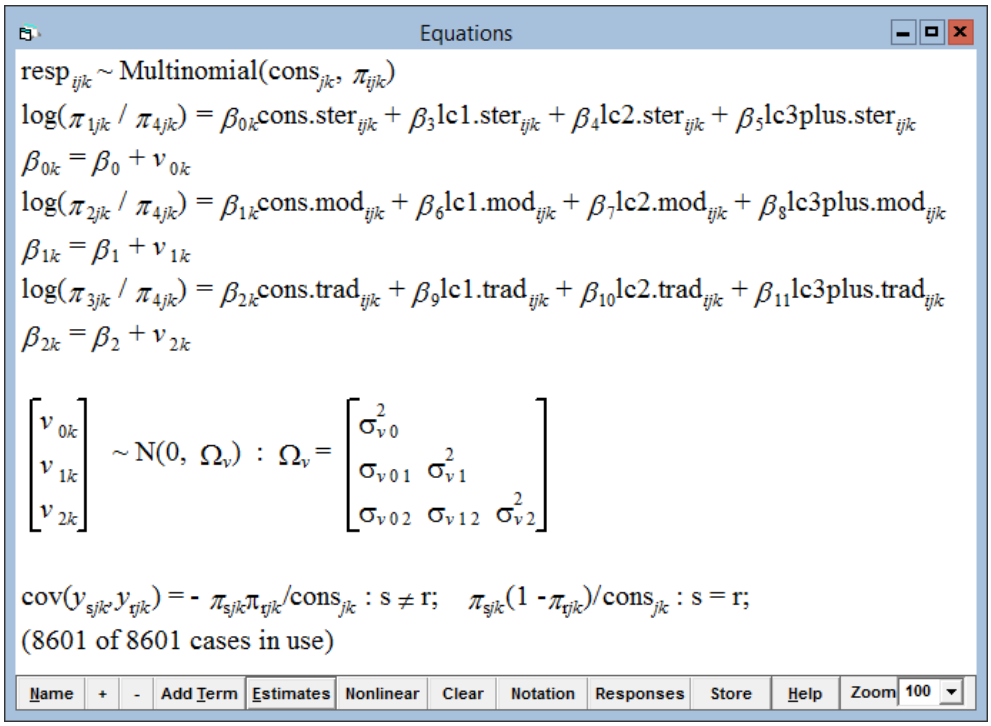
in MLwiN for the estimation of multilevel models that have categorical responses: quasi-likelihood methods (MQL / PQL, 1st or 2nd order) and MCMC methods. See Section 9.2 of this manual and Browne (2003) for further discussion. We shall use the quasi-likelihood methods in this chapter, starting with 1st order MQL and extending to 2nd order PQL on convergence.

10.5 Fitting a two-level random intercept model

To extend the current single-level model to a two-level random intercept model:

- Click on **resp** $_{ij}$, and change **N levels:** from **2-ij** to **3-ijk**
- Next to **level 3(k):**, select **district** from the drop down list and click **Done**
- Now click on **cons.ster** (or its coefficient), check **k(district_long)** in the **X variable** window and click **Done**
- Next, click on **cons.mod**, check **k(district_long)** and click **Done**
- Finally, click on **cons.trad**, check **k(district_long)** and click **Done**

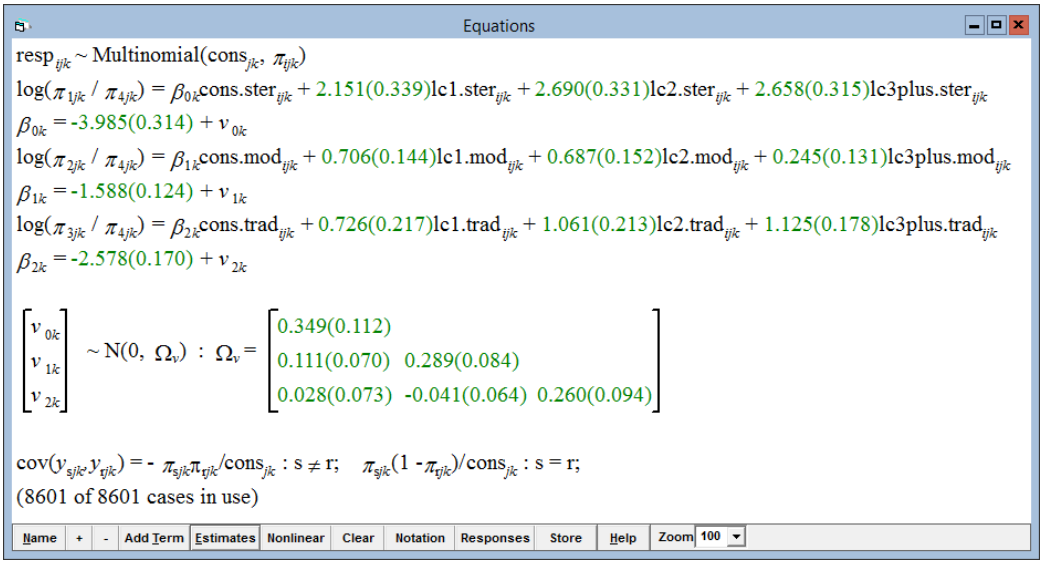
After clicking on **Estimates**, the **Equations** window should look like this:


Equations
$resp_{ijk} \sim Multinomial(cons_{jk}, \pi_{ijk})$
$\log(\pi_{1jk} / \pi_{4jk}) = \beta_{0k} cons.ster_{ijk} + \beta_3 lc1.ster_{ijk} + \beta_4 lc2.ster_{ijk} + \beta_5 lc3plus.ster_{ijk}$
$\beta_{0k} = \beta_0 + v_{0k}$
$\log(\pi_{2jk} / \pi_{4jk}) = \beta_{1k} cons.mod_{ijk} + \beta_6 lc1.mod_{ijk} + \beta_7 lc2.mod_{ijk} + \beta_8 lc3plus.mod_{ijk}$
$\beta_{1k} = \beta_1 + v_{1k}$
$\log(\pi_{3jk} / \pi_{4jk}) = \beta_{2k} cons.trad_{ijk} + \beta_9 lc1.trad_{ijk} + \beta_{10} lc2.trad_{ijk} + \beta_{11} lc3plus.trad_{ijk}$
$\beta_{2k} = \beta_2 + v_{2k}$
$\begin{bmatrix} v_{0k} \\ v_{1k} \\ v_{2k} \end{bmatrix} \sim N(0, \Omega_v) : \Omega_v = \begin{bmatrix} \sigma^2_{v0} & & \\ \sigma_{v01} & \sigma^2_{v1} & \\ \sigma_{v02} & \sigma_{v12} & \sigma^2_{v2} \end{bmatrix}$
$cov(y_{sjk}, y_{rjk}) = - \pi_{sjk}\pi_{rjk}/cons_{jk} : s \neq r; \quad \pi_{sjk}(1 - \pi_{rjk})/cons_{jk} : s = r;$
(8601 of 8601 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

Just as a single-level model was formulated as a two-level model in MLwiN, a two-level model is formulated as a three-level model, hence the three sub-

scripts ijk. The additional k subscript indicates the district. The district-level random effects, denoted by $u_j^{(s)}(s= 1, 2, 3)$ in (10.4), are v_{0k}, v_{1k} and v_{2k} in MLwiN with variance-covariance matrix Ω_v.

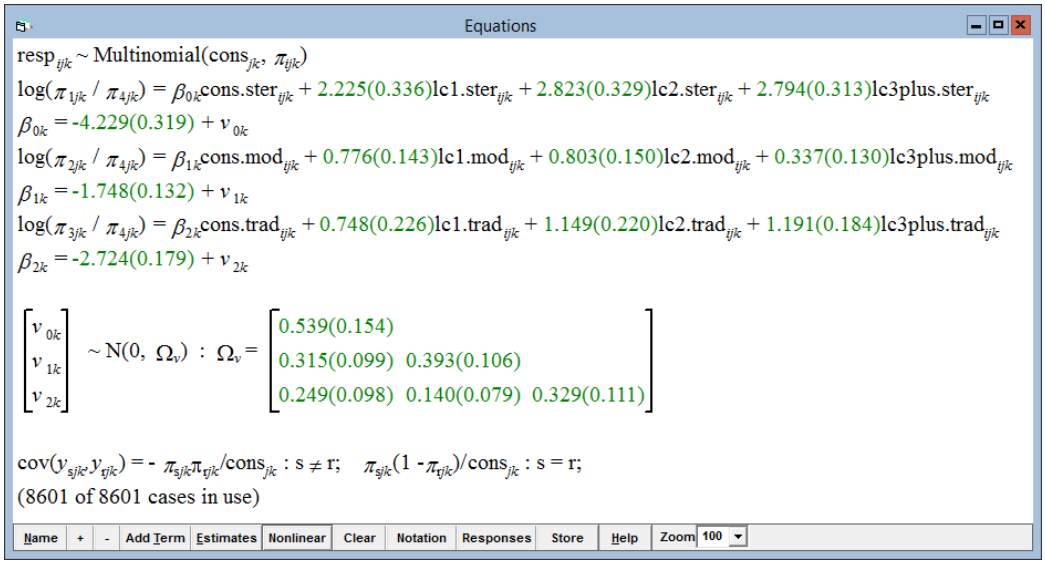
Fit the model using the default 1st order MQL procedure. You should see the following:


Equations
resp_ijk ~ Multinomial(cons_jk, π_ijk)
log(π_1jk / π_4jk) = β_0k cons.ster_ijk + 2.151(0.339)lc1.ster_ijk + 2.690(0.331)lc2.ster_ijk + 2.658(0.315)lc3plus.ster_ijk
β_0k = -3.985(0.314) + v_0k
log(π_2jk / π_4jk) = β_1k cons.mod_ijk + 0.706(0.144)lc1.mod_ijk + 0.687(0.152)lc2.mod_ijk + 0.245(0.131)lc3plus.mod_ijk
β_1k = -1.588(0.124) + v_1k
log(π_3jk / π_4jk) = β_2k cons.trad_ijk + 0.726(0.217)lc1.trad_ijk + 1.061(0.213)lc2.trad_ijk + 1.125(0.178)lc3plus.trad_ijk
β_2k = -2.578(0.170) + v_2k
v_0k, v_1k, v_2k ~ N(0, Ω_v) : Ω_v = 0.349(0.112); 0.111(0.070) 0.289(0.084); 0.028(0.073) -0.041(0.064) 0.260(0.094)
cov(y_sjk, y_rjk) = - π_sjk π_rjk / cons_jk : s ≠ r; π_sjk(1 - π_rjk)/cons_jk : s = r;
(8601 of 8601 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

Now change to 2nd order PQL:

- Click on the **Nonlinear** button
- Change from **MQL** to **PQL**
- Change from **1st order** to **2nd order**
- Click **Done**
- Click **More**

You should get these results:


Equations
resp_ijk ~ Multinomial(cons_jk, π_ijk)
log(π_1jk / π_4jk) = β_0k cons.ster_ijk + 2.225(0.336)lc1.ster_ijk + 2.823(0.329)lc2.ster_ijk + 2.794(0.313)lc3plus.ster_ijk
β_0k = -4.229(0.319) + v_0k
log(π_2jk / π_4jk) = β_1k cons.mod_ijk + 0.776(0.143)lc1.mod_ijk + 0.803(0.150)lc2.mod_ijk + 0.337(0.130)lc3plus.mod_ijk
β_1k = -1.748(0.132) + v_1k
log(π_3jk / π_4jk) = β_2k cons.trad_ijk + 0.748(0.226)lc1.trad_ijk + 1.149(0.220)lc2.trad_ijk + 1.191(0.184)lc3plus.trad_ijk
β_2k = -2.724(0.179) + v_2k
v_0k
v_1k
v_2k
~ N(0, Ω_v) : Ω_v =
0.539(0.154)
0.315(0.099) 0.393(0.106)
0.249(0.098) 0.140(0.079) 0.329(0.111)
cov(y_sjk, y_tjk) = - π_sjk π_tjk/cons_jk : s ≠ r; π_sjk(1 - π_tjk)/cons_jk : s = r;
(8601 of 8601 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

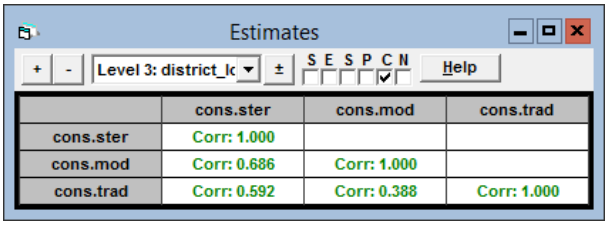
Notice that there are some sizeable differences between the 1st order MQL and 2nd order PQL estimates, particularly for the random part parameters. For multinomial logit models, the 1st order MQL approximation may produce severely biased estimates. Users are advised to use 2nd order PQL or MCMC methods (see Browne (2003); Chapter 20).

In each of the three contrasts, the estimate of district-level variance is large relative to its standard error, suggesting that there is unexplained district-level variation in the use of each type of contraception method. The random effect covariances are all positive, indicating that districts with high (low) use of one type of method also tend to have high (low) use of other methods. It would be easier, however, to interpret the *correlations* rather than the covariances. To obtain the district-level correlations:

- From the **Model** menu, select **Estimate tables**
- Select **Level 3: district_long** from the drop-down list that currently shows **FIXED PART**
- Also at the top of the **Estimates** window, check **C** and uncheck **S**, **E**, **S** and **P**

Note that these checkboxes (S, E, S, P, C, and N) control what is displayed in the table. Click ***Help*** *for more details*

You should see the following correlation matrix:



	cons.ster	cons.mod	cons.trad
cons.ster	Corr: 1.000		
cons.mod	Corr: 0.686	Corr: 1.000	
cons.trad	Corr: 0.592	Corr: 0.388	Corr: 1.000

The highest correlation at the district level is between use of sterilization and use of modern reversible methods, which would be expected since both of these types of method are promoted by family planning programmes. The correlation between use of sterilization and use of traditional methods is also high.

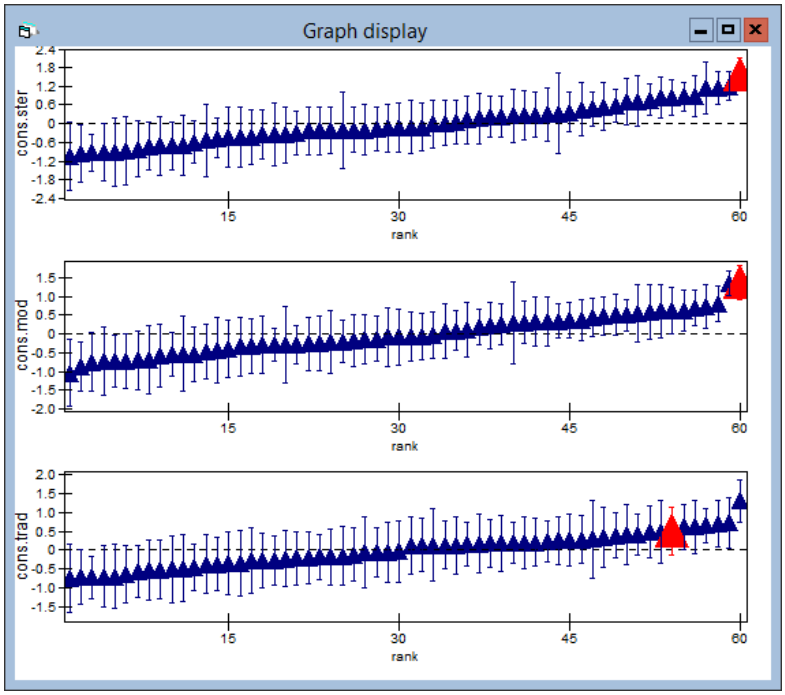
We will now look at residual estimates to further explore the extent of district-level variation and to see if there are any 'outlying' districts with high or low contraceptive prevalence, after adjusting for differences in fertility. To obtain estimates and plots of the three sets of district-level residuals:

- From the **Model** menu, select **Residuals**
- At the bottom of the **Residuals** window, next to **level:**, select **3:district_long**
- Change the **SD(comparative)** multiplier to **1.96** and click **Calc** The residuals corresponding to the contrasts for sterilization, modern and traditional versus none are output to **c300, c301** and **c302**, respectively
- From the **Data Manipulation** menu, select **Names**
- Assign the name **res_ster** to **c300**, **res_mod** to **c301** and **res_trad** to **c302**
- Now return to the **Residuals** window and click on the **Plots** tab
- Select **residual +/− 1.96 sd x rank** and click **Apply**

You should see a figure like the one on the next page containing three 'caterpillar' plots. These show residual estimates with 95% confidence intervals.

To identify districts with particularly large (small) prevalences of use of particular contraception methods, we can highlight them in the plots as follows:

- In the **first plot**, click on the confidence interval for the district with the *largest positive residual*, i.e, the highest prevalence of sterilization.
- The **Identify point** window will appear, informing you that this point corresponds to **district ID 56**.
- Under **In graphs** click on **highlight(style 1)**, then **Apply**.
- The residual estimates for the district with ID=56 will be highlighted in red in all 3 plots. Notice that this district also has the highest prevalence of modern reversible methods and high prevalence of traditional methods.



Repeating the above steps for the district with the largest *negative* residual for sterilization, using a different highlighting style, reveals that the district with ID=11 has the lowest prevalence not only of sterilization but of modern and traditional methods. The tendency for districts to have a similar ranking for all three types of method is reflected in the positive correlations between district random effects.

The next step in the analysis would be to add further explanatory variables and, in particular, to examine whether the district-level indicators of literacy and religiosity can explain district-level variation in use of the different contraceptive methods. We leave this as an exercise for the reader.

Chapter learning outcomes

- ⋆ How to formulate single-level and multilevel multinomial models
- ⋆ How to specify and fit such models in MLwiN
- ⋆ How to interpret the results from such models

Chapter 11

Fitting an Ordered Category Response Model

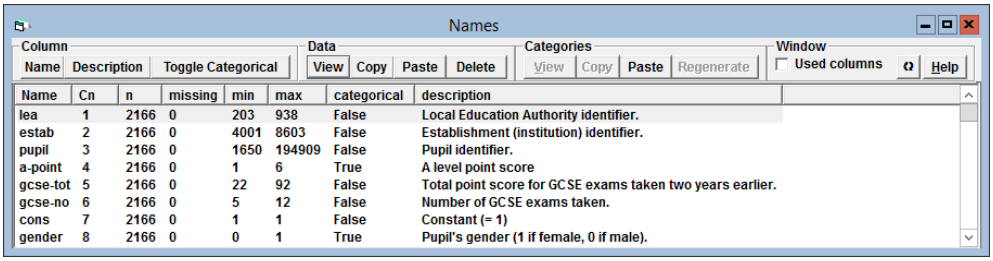
11.1 Introduction

Many kinds of response variables take the form of *ordered* category scales. Attitude measurements, examination grades and disease severity are just a few examples of such variables. Very often in analyses, scores are assigned to the categories, and these scores are treated as if they are measurements on a continuous scale. Typically, however, such scoring systems are arbitrary, and information may be lost or distorted in the conversion. An alternative approach is to retain the categories throughout the analysis. The example analyses presented in this chapter show how this can be done — first using a single-level model, then with a multilevel model. A more detailed discussion of models with ordered categorical responses can be found in Goldstein (2003) and Yang & Woodhouse (2001).

The example data set: chemistry A level grades

The data used in our example are taken from a large data set comprising the results of all A level GCSE examinations in England during the period 1994 to 1997 (Yang & Woodhouse, 2001). For present purposes, we have chosen results for chemistry from one examination board in 1997. We have data from 2166 students in 219 educational institutions.

Open the data file **alevchem.ws**, and you will see the following list of variables:



Name	Cn	n	missing	min	max	categorical	description
lea	1	2166	0	203	938	False	Local Education Authority identifier.
estab	2	2166	0	4001	8603	False	Establishment (institution) identifier.
pupil	3	2166	0	1650	194909	False	Pupil identifier.
a-point	4	2166	0	1	6	True	A level point score
gcse-tot	5	2166	0	22	92	False	Total point score for GCSE exams taken two years earlier.
gcse-no	6	2166	0	5	12	False	Number of GCSE exams taken.
cons	7	2166	0	1	1	False	Constant (= 1)
gender	8	2166	0	0	1	True	Pupil's gender (1 if female, 0 if male).

The variables are defined as follows:

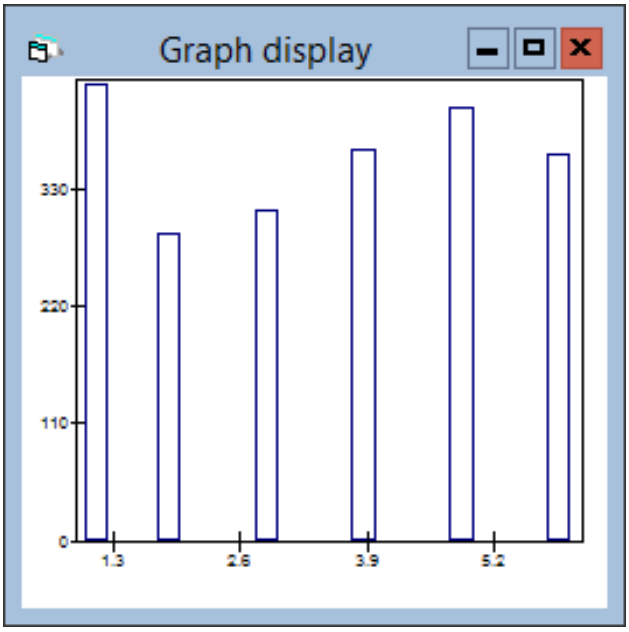
Variable	*Description*
lea	Local Education Authority (not used in this analysis)
estab	Institution identification
pupil	Pupil identification
a-point	A level point score (see below for description)
gcse-tot	Total point score for GCSE exams taken two years earlier
gcse-no	Number of GCSE exams taken
cons	Constant (= 1)
gender	1 if female, 0 if male

The codes in the variable **a-point** correspond to the following grades: 1 = F; 2 = E; 3 = D; 4 = C; 5 = B; and 6 = A. The standard procedure when analysing such examination data is to use a scoring system that assigns a value 0 to grade F, a value 2 to grade E, and finally a value 10 to grade A and then treats this as a continuous response variable.

11.2 An analysis using the traditional approach

To provide a point of comparison for the categorical response models, we will begin our series of analyses by fitting a single-level model that treats a transformed form of the response variable as if it were continuous.

We can use MLwiN's **Customised graph** window to create a histogram of **a-point**:



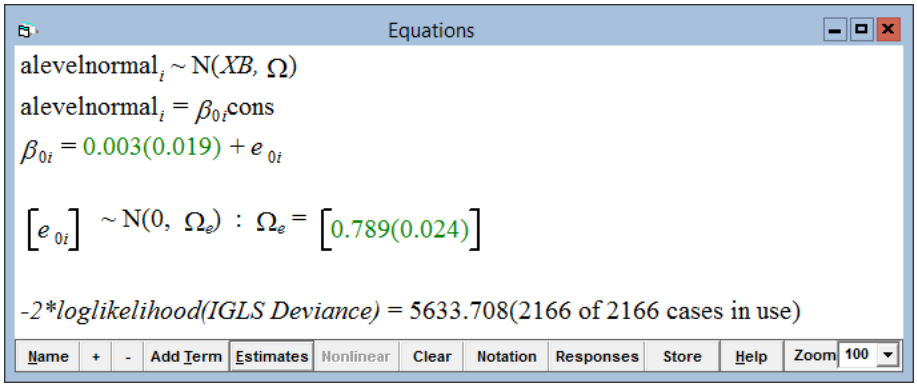
From this we see that the distribution of our response variable is certainly not Normal. In view of this non-normality, we shall make a transformation to Normal scores. To each response category, this transformation assigns the value from the inverse of the standard (0,1) Normal cumulative distribution for the estimated proportion of pupils from the response variable's original distribution. See Darlington (1997) for further details and examples.

You can use MLwiN's **NSCO** command to create a new response variable of Normal scores.

- Select **Command interface** from the **Data Manipulation** menu
- In the bottom box of the **Command interface** window, type:

```
► NSCO 'a-point' c12
► NAME c12 'alevelnormal'
```

We will treat **alevelnormal** as a continuous response variable. We first fit the simplest possible single-level model involving just an intercept term. To do this, use the **Equations** window to define the response variable as **alevelnormal** (with a Normal error distribution), and set up a single-level model, using **pupil** as the level 1 identifier. Add the variable **cons** as the explanatory variable. We obtain the following estimates:


Equations
alevelnormal$_i$ ~ N(XB, Ω)
alevelnormal$_i$ = β_{0i}cons
β_{0i} = 0.003(0.019) + e_{0i}
[e_{0i}] ~ N(0, Ω_e) : Ω_e = [0.789(0.024)]
-2*loglikelihood(IGLS Deviance) = 5633.708(2166 of 2166 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

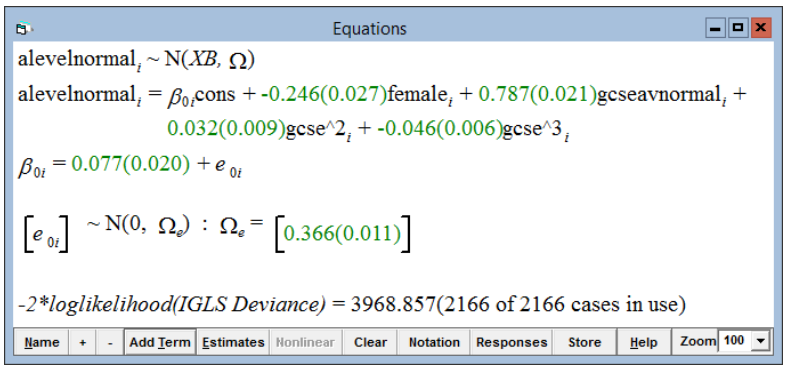
In subsequent models we shall include each pupil's average GCSE score as an explanatory variable. We first compute the mean then Normalise it. To allow for the possibility that the relationship is non-linear, we create new variables equal to the square and cube of the mean.

- In the **Command interface** window, enter the following:

```
calc c9=c5/c6
nsco c9 c9
calc c10=c9^2
calc c11=c9^3
name c9 'gcseavnormal' c10 'gcse^2' c11 'gcse^3'
```

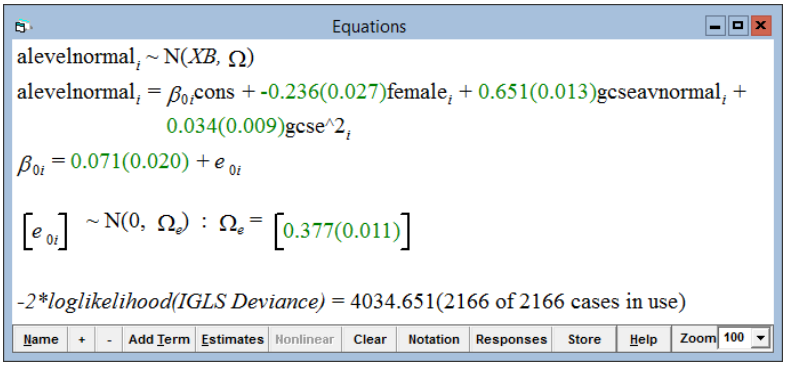
You can of course do the arithmetic calculations in the **Calculate** window, but the Normal score transformation can only be done using the **NSCO** command.

Now add the newly created GCSE variables and **gender** as explanatory variables and fit the model. You will see the following results:


Equations
alevelnormal$_i$ ~ N(XB, Ω)
alevelnormal$_i$ = β_{0i}cons + -0.246(0.027)female$_i$ + 0.787(0.021)gcseavnormal$_i$ +
0.032(0.009)gcse^2$_i$ + -0.046(0.006)gcse^3$_i$
β_{0i} = 0.077(0.020) + e_{0i}
[e_{0i}] ~ N(0, Ω_e) : Ω_e = [0.366(0.011)]
-2*loglikelihood(IGLS Deviance) = 3968.857(2166 of 2166 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

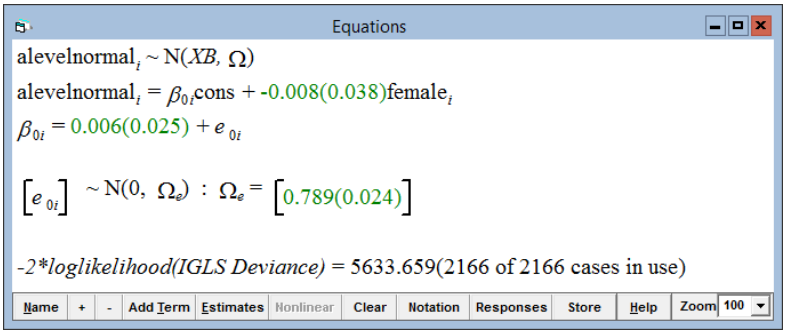
The coefficients for the quadratic and cubic terms are both significant, but for simplicity we shall ignore the cubic term in the following analyses. When we fit just the linear and quadratic GCSE terms, their coefficients change

very little.

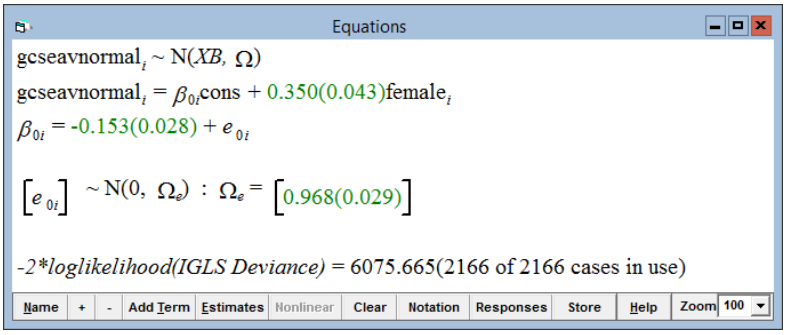

Equations
alevelnormal$_i$ ~ N(XB, Ω)
alevelnormal$_i$ = β_{0i}cons + -0.236(0.027)female$_i$ + 0.651(0.013)gcseavnormal$_i$ +
0.034(0.009)gcse^2$_i$
β_{0i} = 0.071(0.020) + e_{0i}
[e_{0i}] ~ N(0, Ω_e) : Ω_e = [0.377(0.011)]
-2*loglikelihood(IGLS Deviance) = 4034.651(2166 of 2166 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

Note also that most of the level 1 variance (initially 1.0 as a result of the Normalisation) has been explained by the model. We also see a significant gender effect with girls performing worse than boys on average after adjusting for GCSE.

If we do not adjust for GCSE, we obtain the following:


Equations
alevelnormal$_i$ ~ N(XB, Ω)
alevelnormal$_i$ = β_{0i}cons + -0.008(0.038)female$_i$
β_{0i} = 0.006(0.025) + e_{0i}
[e_{0i}] ~ N(0, Ω_e) : Ω_e = [0.789(0.024)]
-2*loglikelihood(IGLS Deviance) = 5633.659(2166 of 2166 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

There is now no significant difference between boys and girls. If we model the GCSE score to examine gender differences, we obtain:


Equations
gcseavnormal$_i$ ~ N(XB, Ω)
gcseavnormal$_i$ = β_{0i}cons + 0.350(0.043)female$_i$
β_{0i} = -0.153(0.028) + e_{0i}
[e_{0i}] ~ N(0, Ω_e) : Ω_e = [0.968(0.029)]
-2*loglikelihood(IGLS Deviance) = 6075.665(2166 of 2166 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

We see that the girls have a higher average GCSE score than boys. Returning to our last model for **alevelnormal** scores (with no adjustment for GCSE), we interpret the absence of a gender difference as simply a reflection of the fact that girls who take Chemistry A levels have a higher prior (GCSE) achievement but make less progress between GCSE and A level.

11.3 A single-level model with an ordered categorical response variable

We now look at a richer model that retains the response variable's original grade categories. Using notation similar to that of Chapter 10, we specify that our original response variable has t categories, indexed by s ($s = 1, \ldots, t$) and that category t is chosen as the reference category. Suppose that the probability of pupil i having a response variable value of s is $\pi_i^{(s)}$.

To exploit the ordering we base our models upon the *cumulative* response probabilities rather than the response probabilities for each separate category. We define the cumulative response probabilities as

$$E(y_i^{(s)}) = \gamma_i^{(s)} = \sum_{h=1}^{s} \pi_i^{(h)}, \quad s = 1, \ldots, t-1 \tag{11.1}$$

Here, $y_i^{(s)}$ are the observed cumulative proportions (out of a total n_i observations-one in our example) for the ith pupil. Expressing the category probabilities in terms of the cumulative probabilities we have:

$$\begin{aligned} \pi_i^{(h)} &= \gamma_i^{(h)} - \gamma_i^{(h-1)}, \quad 1 < h < t \\ \pi_i^{(1)} &= \gamma_i^{(1)}; \quad \gamma_i^{(t)} = 1 \end{aligned} \tag{11.2}$$

A common model choice is the *proportional odds* model with a logit link, namely:

$$\begin{aligned} &\gamma_i^{(s)} = \{1 + \exp -[\alpha^{(s)} + (X\beta)_i]\}^{-1} \\ &\text{or} \\ &\text{logit}(\gamma_i^{(s)}) = \alpha^{(s)} + (X\beta)_i \end{aligned} \tag{11.3}$$

This implies that increasing values of the linear component are associated with increasing probabilities as s increases.

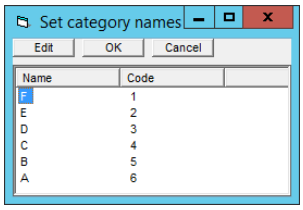
If we assume an underlying multinomial distribution for the category probabilities, the cumulative proportions have a covariance matrix given by

$$\text{cov}(y_i^{(s)}, y_i^{(r)}) = \gamma_i^{(s)}(1 - \gamma_i^{(r)})/n_i, \quad s \leq r \tag{11.4}$$

We can fit models to these cumulative proportions (or counts conditional on a fixed total) in the same way as with a regular multinomial response vector, substituting this covariance matrix. (For a discussion of fitting the standard *unordered* multinomial, see Chapter 10.)

We now look at models that directly fit the ordered grade categories using the model described above. Start by looking at the **a-point** variable.

- Click on **a-point** in the **Names** window
- Click on the **View** button in the categories section



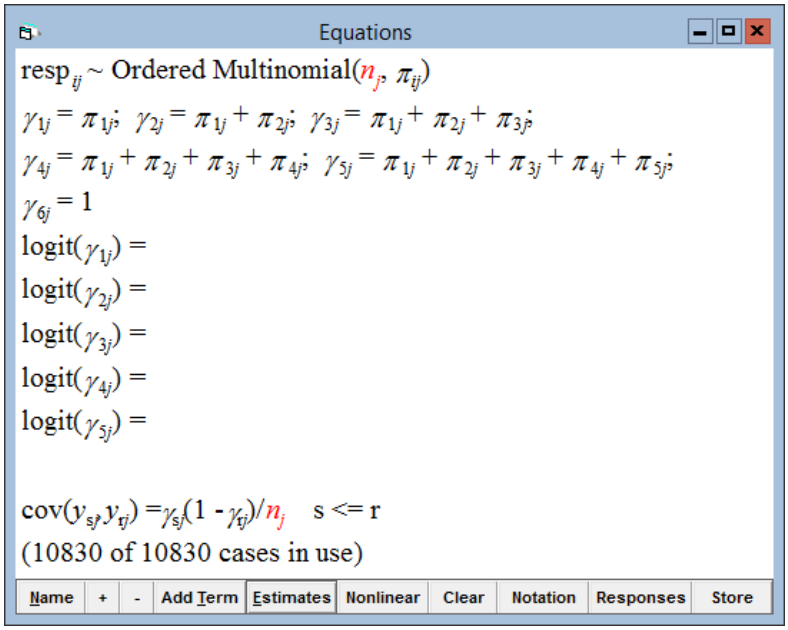
We see that **a-point** has already been defined as a categorical variable

- Click **OK**

Now set up the model.

- From the **Model** menu, select **Equations**
- Click **Clear**
- In the **Equations** window, click on the **Name** button
- Click on y, and make the following selections in the **Y variable** window:
  y: **a-point**
  N levels: **1-i**
  level 1(i): **pupil**
- Click **done**
- Click on the N in the **Equations** window, scroll down and select **Multinomial**
- Under **Multinomial options**, select **Ordered proportional odds**
- Use the default **logit** link function
- Next to **ref category** select **A** from the drop-down list
- Click **Done**
- Click **Estimates**

The **Equations** window now shows the following:


Equations
resp_ij ~ Ordered Multinomial(n_j, π_ij)
γ1j = π1j; γ2j = π1j + π2j; γ3j = π1j + π2j + π3j;
γ4j = π1j + π2j + π3j + π4j; γ5j = π1j + π2j + π3j + π4j + π5j;
γ6j = 1
logit(γ1j) =
logit(γ2j) =
logit(γ3j) =
logit(γ4j) =
logit(γ5j) =
cov(ysj, ytj) =γsj(1 - γtj)/nj s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

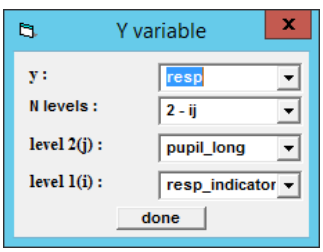
The model is expressed in a form that is similar — though not identical — to the one presented in equations (11.1) to (11.4).

> *Note that y has been replaced by* ***resp****, and that two subscripts are used on this and the π and γ terms. MLwiN has created a two-level formulation of our single-level model in a way that parallels what it did with the unordered category model in Section 10.3. Each pupil, now a level 2 unit, has five response variables (level 1 units)*

The five equations for these response variables are incomplete because we have not yet selected the explanatory variables to include in the model. Referring back to definitions of our categories, we would interpret $\text{logit}(\gamma_{5j})$ as the logit of the expected probability that pupil j had a chemistry grade of B or lower.

If we look at the **Names** window, we will see that several new variables have appeared in vacant columns, e.g., **resp_indicator** and **pupil_long**. The suffix **_long** is created by MLwiN to distinguish each new variable created automatically in an expanded data set. Each of these new columns has a length of 10830 ($= 5 \times 2166$) because there are 5 responses per pupil.

If we click on **resp** in the **Equations window** we obtain:



The double subscripting and new identification code variables further illustrate that the model has become a 2-level model with the response category as level 1.

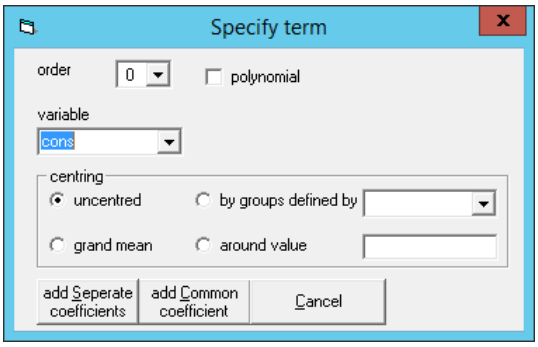
Now we need to define the denominator vector. For each pupil only one grade is possible, so the value of n_j is always 1. Thus, we need to tell MLwiN to use a column of 1s as the denominator vector. We already have cons, so we can use that.

- Click on the term n_j in red and choose **cons**

We can now start adding explanatory variables.

- Click on the **Add term** button and select **cons**

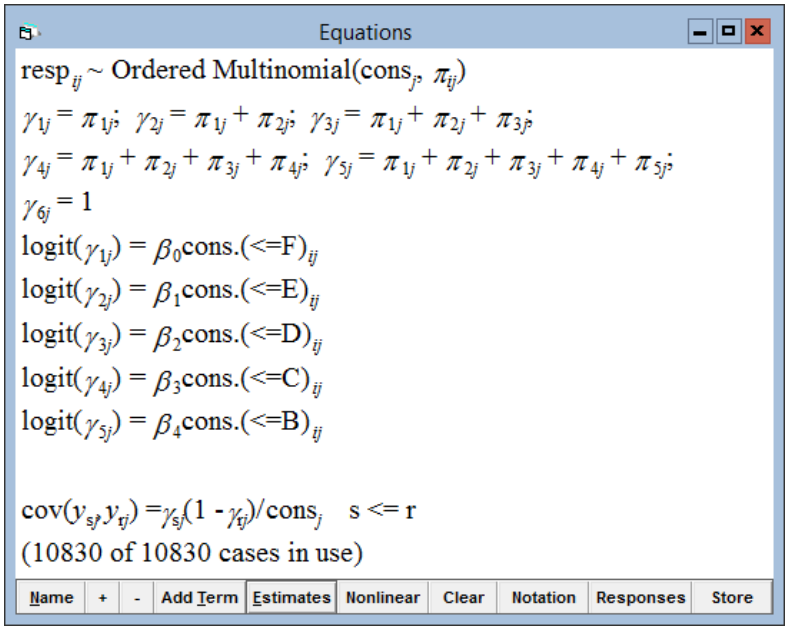
We obtain the following window:



We need to decide whether to fit a separate intercept for each of the five response variables or to use a common intercept coefficient. We begin by choosing the former option.

- Click on the add Separate coefficients button

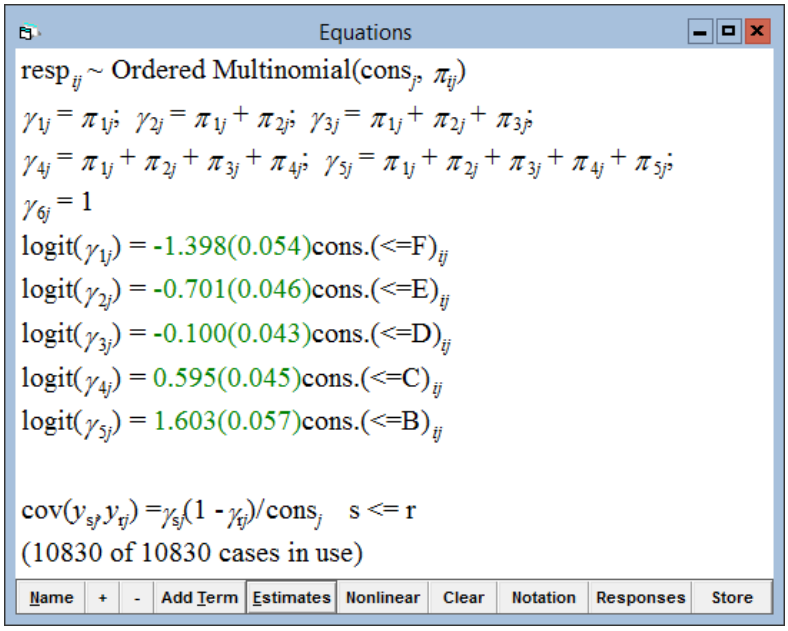
The **Equations** window now shows:


Equations
resp$_{ij}$ ~ Ordered Multinomial(cons$_j$, π_{ij})
$\gamma_{1j} = \pi_{1j}$; $\gamma_{2j} = \pi_{1j} + \pi_{2j}$; $\gamma_{3j} = \pi_{1j} + \pi_{2j} + \pi_{3j}$;
$\gamma_{4j} = \pi_{1j} + \pi_{2j} + \pi_{3j} + \pi_{4j}$; $\gamma_{5j} = \pi_{1j} + \pi_{2j} + \pi_{3j} + \pi_{4j} + \pi_{5j}$;
$\gamma_{6j} = 1$
logit(γ_{1j}) = $\beta_0$cons.(<=F)$_{ij}$
logit(γ_{2j}) = $\beta_1$cons.(<=E)$_{ij}$
logit(γ_{3j}) = $\beta_2$cons.(<=D)$_{ij}$
logit(γ_{4j}) = $\beta_3$cons.(<=C)$_{ij}$
logit(γ_{5j}) = $\beta_4$cons.(<=B)$_{ij}$
cov(y_{sj},y_{tj}) =γ_{sj}(1 - γ_{tj})/cons$_j$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The naming of the explanatory variables indicates that we are fitting an ordered proportional odds model as given in equations (11.1) to (11.4). We will now run this model, which simply fits a separate intercept for each grade.

- In the Equations window, click the **Nonlinear** button
- In the Nonlinear Estimation window, click on **Use Defaults**
- Click **Done**
- Click **Start**

We obtain the following:


Equations
resp$_{ij}$ ~ Ordered Multinomial(cons$_j$, π_{ij})
$\gamma_{1j} = \pi_{1j}$; $\gamma_{2j} = \pi_{1j} + \pi_{2j}$; $\gamma_{3j} = \pi_{1j} + \pi_{2j} + \pi_{3j}$;
$\gamma_{4j} = \pi_{1j} + \pi_{2j} + \pi_{3j} + \pi_{4j}$; $\gamma_{5j} = \pi_{1j} + \pi_{2j} + \pi_{3j} + \pi_{4j} + \pi_{5j}$;
$\gamma_{6j} = 1$
logit(γ_{1j}) = -1.398(0.054)cons.(<=F)$_{ij}$
logit(γ_{2j}) = -0.701(0.046)cons.(<=E)$_{ij}$
logit(γ_{3j}) = -0.100(0.043)cons.(<=D)$_{ij}$
logit(γ_{4j}) = 0.595(0.045)cons.(<=C)$_{ij}$
logit(γ_{5j}) = 1.603(0.057)cons.(<=B)$_{ij}$
cov(y_{sj},y_{tj}) =γ_{sj}(1 - γ_{tj})/cons$_j$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

If we take the antilogit of the first coefficient (-1.398), we obtain 0.198, the

estimated probability that a pupil's chemistry grade is F. The estimates from this simple model agree with proportions we can calculate directly from the data using the **Tabulate** window. The proportion of pupils with a grade of F is 19.8%. The probability that a pupil has a grade of F or E is given by the antilogit of -0.701, i.e, 0.332, and the proportion of pupils with either of these grades is 33.1% (as we noted earlier). We shall look at interpretations in more detail later, but note for now that this model is providing more detailed information than was provided by the continuous response model. The latter just averaged grade scores.

11.4 A two-level model

The two-level ordered category response model is a generalisation of the single-level model, as shown in the following set of corresponding model equations:

$$\begin{aligned} E(y_{ij}^{(s)}) = \gamma_{ij}^{(s)} &= \sum_{h=1}^{s} \pi_{ij}^{(h)}, \qquad s = 1, ..., t-1 \\ \mathrm{cov}(y_{ij}^{(s)}, y_{ij}^{(r)}) &= \gamma_{ij}^{(s)}(1 - \gamma_{ij}^{(r)})/n_{ij}, \quad s \leq r \\ \gamma_{ij}^{(s)} &= \{1 + \exp -[\alpha^{(s)} + (X\beta)_{ij} + Z_{ij}u_j]\}^{-1} \end{aligned} \tag{11.5}$$

or

$$\begin{aligned} \mathrm{logit}(\gamma_{ij}^{(s)}) &= \alpha^{(s)} + (X\beta)_{ij} + Z_{ij}u_j \\ \pi_{ij}^{(h)} &= \gamma_{ij}^{(h)} - \gamma_{ij}^{(h-1)}, \quad 1 < h < t \\ \pi_{ij}^{(1)} &= \gamma_{ij}^{(1)}; \quad \gamma_i j^{(t)} = 1 \end{aligned}$$

As we would expect, when fitting this model, MLwiN creates a three-level formulation. We now add educational institution as a third (highest) level in the model we have just fitted.

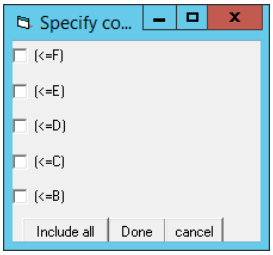
- In the **Equations** window, click on **resp**
- Beside **N levels**: in the **Y variable** window, select **3-ijk**
- Beside **level 3(k)**:, select **estab**
- Click **done**

Note in passing that MLwiN has created a new column, ***estab_long*** *to serve as the actual identifier variable used during fitting. Using the* ***Names*** *window, you can examine this; the five intercept variables derived from* ***cons****; and the full* ***denom*** *variable.*

We now need to define the variation at institution level. One possibility is to allow each category's intercept term to vary, giving us a 5 x 5 covariance matrix at level 3. To do this we would simply click on each **cons.(<=*)** term in turn and in the **X variable** widow, check the **k(estab_long)** box. If we did this, however, we would essentially be fitting a simple multinomial two-level model, which also has a 5 x 5 covariance matrix (see Chapter 10). Instead we will fit a single variance term at the institution level.

- Click on the **Add Term** button on the **Equations** window toolbar
- In the **variable** box of the **Specify term** window, select **cons**
- Click the **add Common coefficient** button

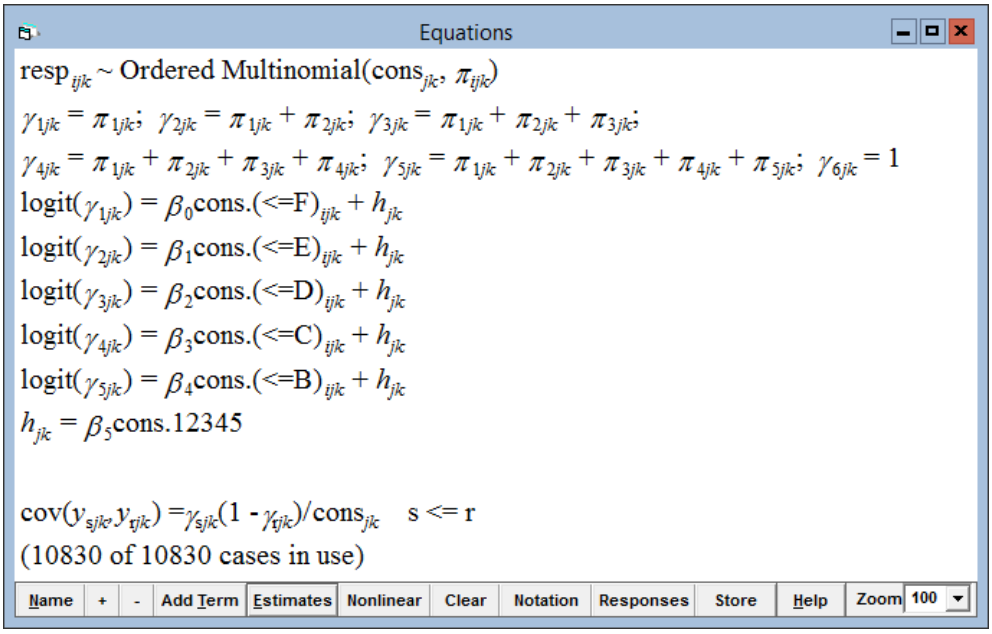
The following window appears:



The terms entered in the model that correspond to the boxes you check in the **Specify common pattern** window will all have equal coefficients. So, to fit our common level 3 variance:

- Click **Include all**
- Click **Done**

The **Equations** window now shows:

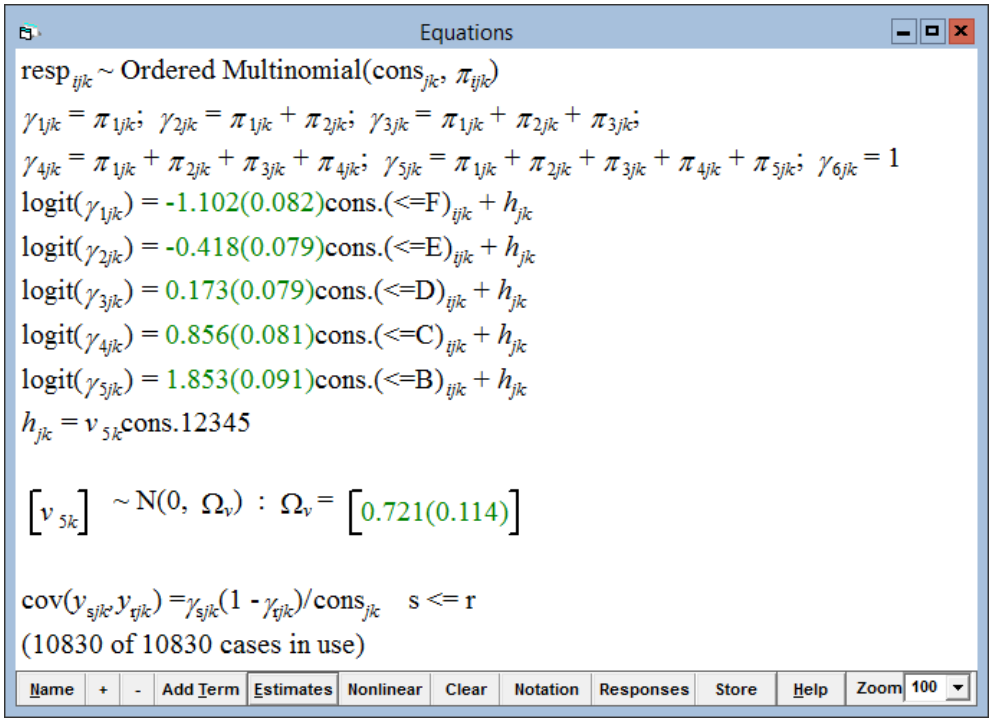

Equations
resp$_{ijk}$ ~ Ordered Multinomial(cons$_{jk}$, π_{ijk})
$\gamma_{1jk} = \pi_{1jk}$; $\gamma_{2jk} = \pi_{1jk} + \pi_{2jk}$; $\gamma_{3jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk}$;
$\gamma_{4jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk}$; $\gamma_{5jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk} + \pi_{5jk}$; $\gamma_{6jk} = 1$
logit(γ_{1jk}) = $\beta_0$cons.(<=F)$_{ijk}$ + h_{jk}
logit(γ_{2jk}) = $\beta_1$cons.(<=E)$_{ijk}$ + h_{jk}
logit(γ_{3jk}) = $\beta_2$cons.(<=D)$_{ijk}$ + h_{jk}
logit(γ_{4jk}) = $\beta_3$cons.(<=C)$_{ijk}$ + h_{jk}
logit(γ_{5jk}) = $\beta_4$cons.(<=B)$_{ijk}$ + h_{jk}
$h_{jk} = \beta_5$cons.12345
cov(y_{sjk},y_{tjk}) =γ_{sjk}(1 - γ_{tjk})/cons$_{jk}$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

A term h_{jk} has been added to the equation for each response category. This represents terms common to the set of equations, and it is defined immediately following the equations for the response categories. In our example, it consists of the single common coefficient associated with a newly created variable named **cons.12345**. If we had specified a pattern that included only response categories 1 and 3, the new variable would have been named **cons.13**, and its term would have appeared only in the equations for these two response categories. As you can see this procedure allows complete flexibility in specifying patterns of shared coefficients across response categories.

At the moment the model is over-parameterised with a unique coefficient for every response category as well as a common coefficient. We want to use the coefficient of the common variable, **cons.12345**, only to specify a common between-institution variability; we do not need this variable in the fixed part of the model. To specify how the variable is used:

- Click on the **cons.12345** terms
- In the **X variable** window, uncheck the **Fixed Parameter** check box
- Check the **k(estab_long)** check box
- Click **Done**

Check that the default estimation procedure has been selected (first order MQL), and run the first estimation. You should obtain the following results:


Equations
resp$_{ijk}$ ~ Ordered Multinomial(cons$_{jk}$, π_{ijk})
$\gamma_{1jk} = \pi_{1jk}$; $\gamma_{2jk} = \pi_{1jk} + \pi_{2jk}$; $\gamma_{3jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk}$;
$\gamma_{4jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk}$; $\gamma_{5jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk} + \pi_{5jk}$; $\gamma_{6jk} = 1$
logit(γ_{1jk}) = -1.102(0.082)cons.(<=F)$_{ijk}$ + h_{jk}
logit(γ_{2jk}) = -0.418(0.079)cons.(<=E)$_{ijk}$ + h_{jk}
logit(γ_{3jk}) = 0.173(0.079)cons.(<=D)$_{ijk}$ + h_{jk}
logit(γ_{4jk}) = 0.856(0.081)cons.(<=C)$_{ijk}$ + h_{jk}
logit(γ_{5jk}) = 1.853(0.091)cons.(<=B)$_{ijk}$ + h_{jk}
$h_{jk} = v_{5k}$cons.12345
$\begin{bmatrix} v_{5k} \end{bmatrix}$ ~ N(0, Ω_v) : Ω_v = $\begin{bmatrix} 0.721(0.114) \end{bmatrix}$
cov(y_{sjk},y_{tjk}) = γ_{sjk}(1 - γ_{tjk})/cons$_{jk}$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

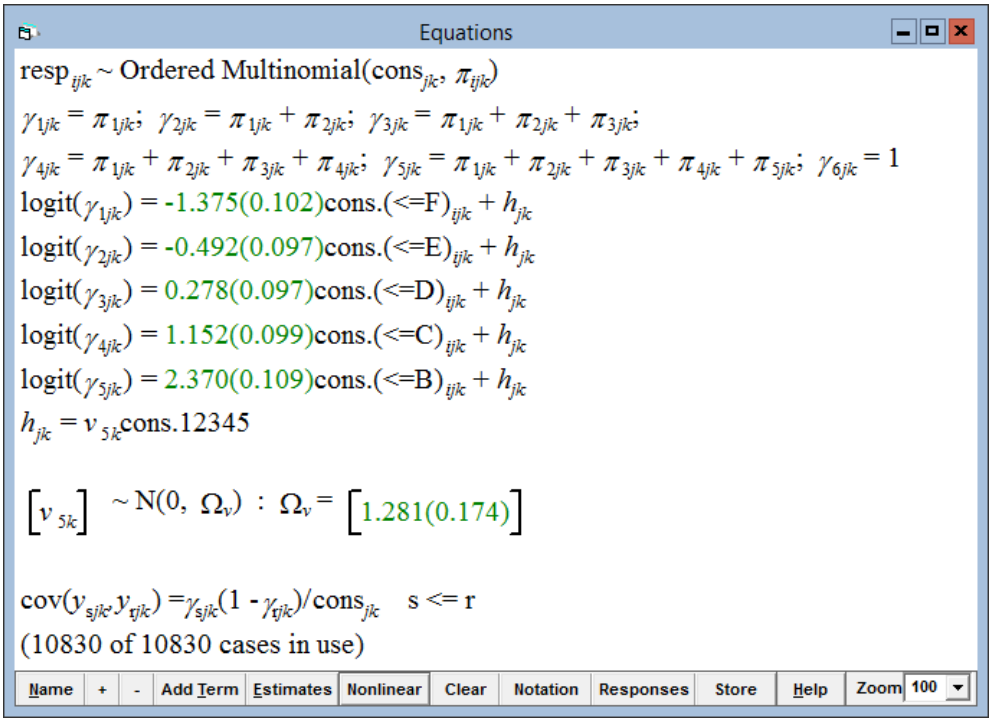
Let us review what we have done so far. The second and third lines specify the cumulative category model. This is followed by five response variable equations, one for each cumulative category. The first explanatory variable in each case is a constant, allowing the intercept to be different for each, as indeed they appear to be. The other explanatory variable, **cons.12345** is also a constant (= 1) whose sole contribution to the model — via its random coefficient — is to add the same random error term to each of the five categories' equations. A common institution — level variance is thus estimated for each category.

Now switch to the preferable method of estimation for this model — second order PQL:

- Click on the **Nonlinear** button
- In the **Nonlinear Estimation** window, select **2nd Order** and **PQL**
- Click **Done**
- Click **Start**

Note that when switching estimation methods and also sometimes when adding new variables you may not be able to proceed by clicking ***More****. Click* ***Start*** *instead.*

The following window shows a large difference between these PQL estimates and the earlier ones; this suggests that the first order MQL procedure underestimates the parameters. We could also get good estimates using MCMC (see Browne (2003)).


Equations
resp$_{ijk}$ ~ Ordered Multinomial(cons$_{jk}$, π_{ijk})
$\gamma_{1jk} = \pi_{1jk}$; $\gamma_{2jk} = \pi_{1jk} + \pi_{2jk}$; $\gamma_{3jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk}$;
$\gamma_{4jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk}$; $\gamma_{5jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk} + \pi_{5jk}$; $\gamma_{6jk} = 1$
logit(γ_{1jk}) = -1.375(0.102)cons.(<=F)$_{ijk}$ + h_{jk}
logit(γ_{2jk}) = -0.492(0.097)cons.(<=E)$_{ijk}$ + h_{jk}
logit(γ_{3jk}) = 0.278(0.097)cons.(<=D)$_{ijk}$ + h_{jk}
logit(γ_{4jk}) = 1.152(0.099)cons.(<=C)$_{ijk}$ + h_{jk}
logit(γ_{5jk}) = 2.370(0.109)cons.(<=B)$_{ijk}$ + h_{jk}
$h_{jk} = v_{5k}$cons.12345
$\begin{bmatrix} v_{5k} \end{bmatrix}$ ~ N(0, Ω_v) : $\Omega_v = \begin{bmatrix} 1.281(0.174) \end{bmatrix}$
cov(y_{sjk},y_{tjk}) = γ_{sjk}(1 - γ_{tjk})/cons$_{jk}$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

Let us now add the GSCE normalised score as an explanatory variable, using a common coefficient across response categories:

- Click on **Add Term**
- In the **variable** box of the **Specify term** window, select **gcseavnormal**
- Click on **add Common coefficient**
- In the **Specify common pattern** window, click on **Include all** and then **Done**

We could have chosen to use **add Separate coefficients** to allow separate coefficients for each category, but this would be formally equivalent to fitting an ordinary multinomial model (see Chapter 10). The important point here is that we are taking advantage of the ordering in the categories to simplify the model structure.

The results of this fit (using PQL2) are shown in the following figure.

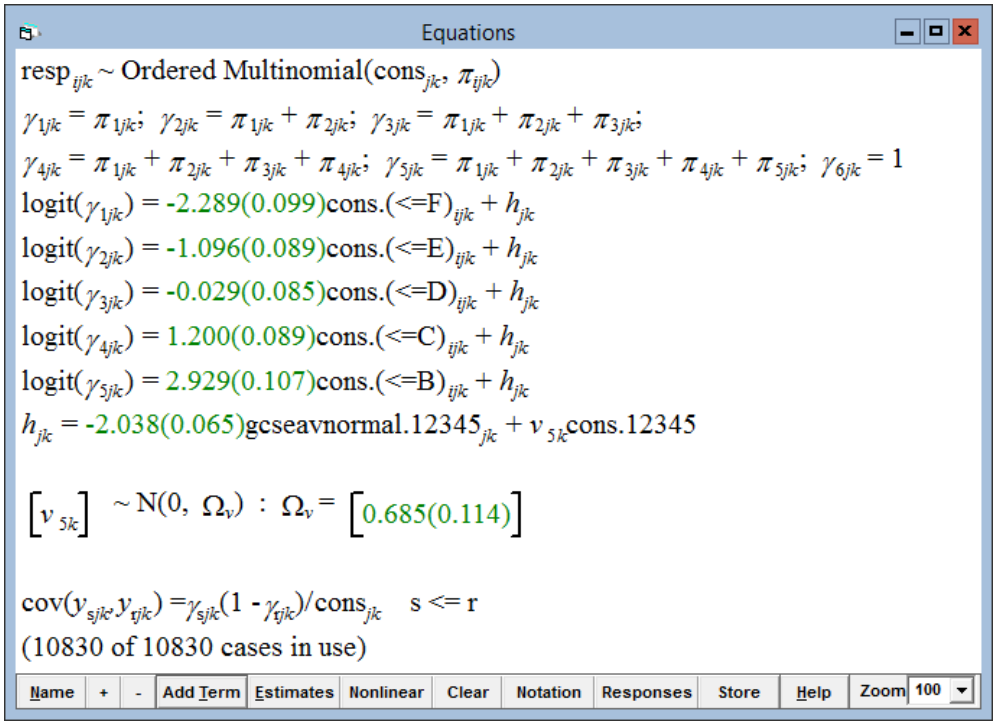
Note how the institution-level variance is reduced considerably when we adjust for the GCSE score.

Note also that since we have treated the highest grade as the reference category, the coefficient of GCSE is negative. That is, as the GCSE score increases, the probability of being in the other (lower) grade categories

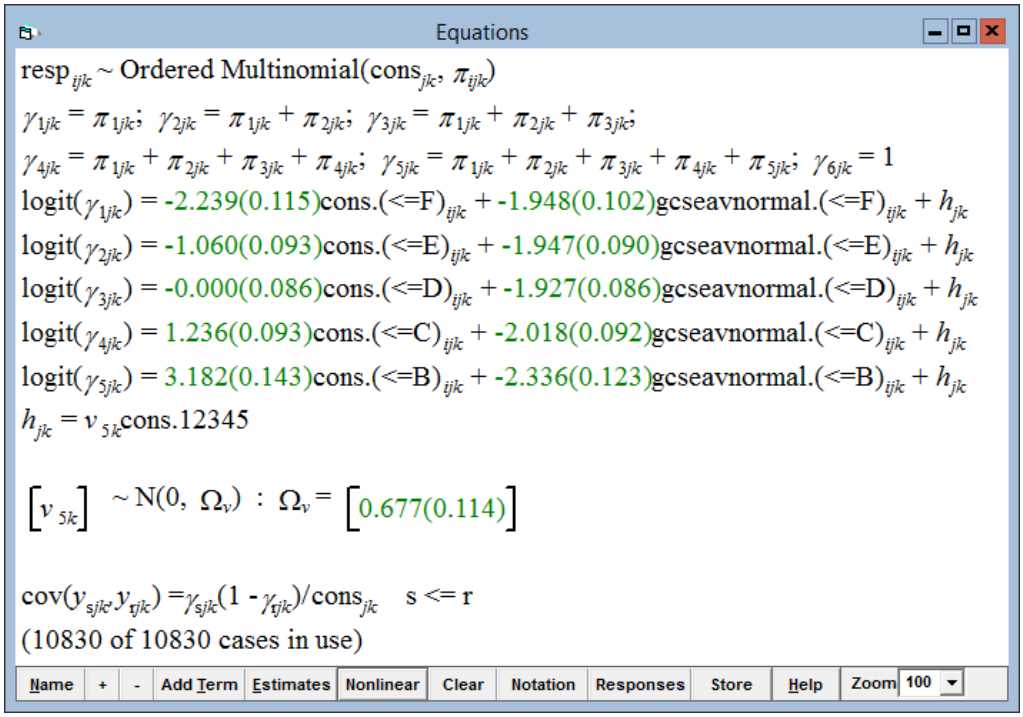
decreases. Thus, for example, the fixed part of the last line of the fitted model specifies:

$$logit(\text{prob of having grade B or below}) = 2.93 - 2.04 \times GCSE\ score$$

so that as the GCSE score increases, the probability of obtaining a chemistry grade of B or lower decreases, or equivalently the probability of a grade of A increases.


Equations
resp$_{ijk}$ ~ Ordered Multinomial(cons$_{jk}$, π_{ijk})
$\gamma_{1jk} = \pi_{1jk}$; $\gamma_{2jk} = \pi_{1jk} + \pi_{2jk}$; $\gamma_{3jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk}$;
$\gamma_{4jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk}$; $\gamma_{5jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk} + \pi_{5jk}$; $\gamma_{6jk} = 1$
logit(γ_{1jk}) = -2.289(0.099)cons.(<=F)$_{ijk}$ + h_{jk}
logit(γ_{2jk}) = -1.096(0.089)cons.(<=E)$_{ijk}$ + h_{jk}
logit(γ_{3jk}) = -0.029(0.085)cons.(<=D)$_{ijk}$ + h_{jk}
logit(γ_{4jk}) = 1.200(0.089)cons.(<=C)$_{ijk}$ + h_{jk}
logit(γ_{5jk}) = 2.929(0.107)cons.(<=B)$_{ijk}$ + h_{jk}
h_{jk} = -2.038(0.065)gcseavnormal.12345$_{jk}$ + v_{5k}cons.12345
$[v_{5k}]$ ~ N(0, Ω_v) : Ω_v = [0.685(0.114)]
cov(y_{sjk},y_{tjk}) = γ_{sjk}(1 - γ_{tjk})/cons$_{jk}$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

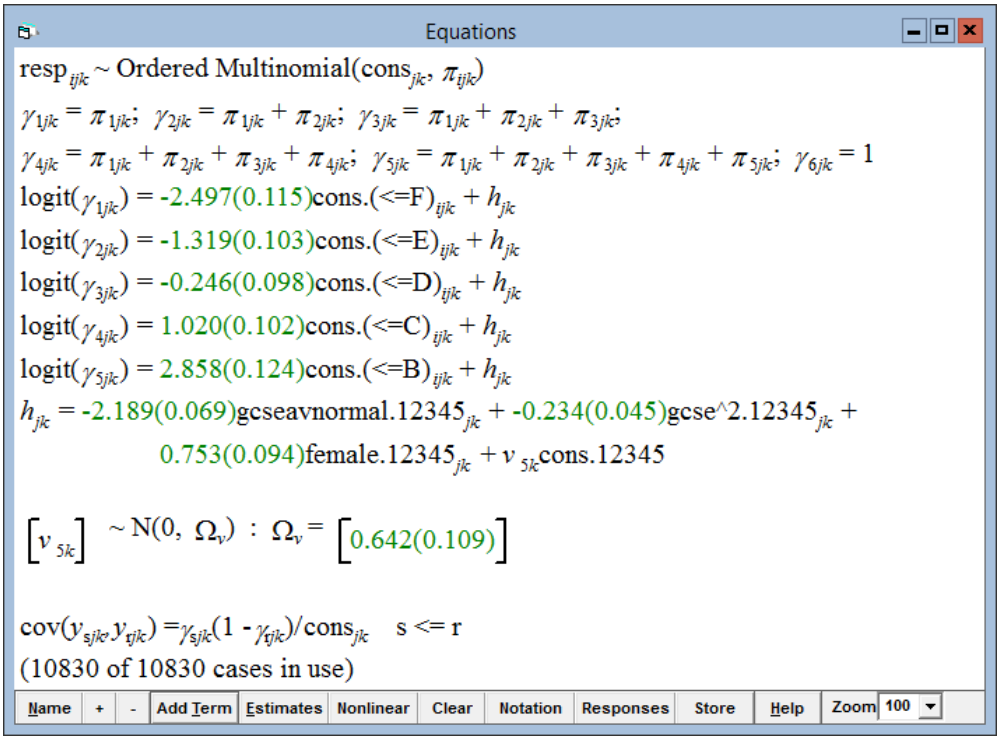
Let's see the effect of allowing each response category equation to have its own coefficient for GCSE. To do this, delete the common GCSE term (**gcseavnormal.12345**) and add **gcseavnormal** again, this time using the **add Separate coefficients** button. Fit the model again using PQL2, and ignore any 'numerical warnings' that appear. We get the following:


Equations
resp$_{ijk}$ ~ Ordered Multinomial(cons$_{jk}$, π_{ijk})
$\gamma_{1jk} = \pi_{1jk}$; $\gamma_{2jk} = \pi_{1jk} + \pi_{2jk}$; $\gamma_{3jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk}$;
$\gamma_{4jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk}$; $\gamma_{5jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk} + \pi_{5jk}$; $\gamma_{6jk} = 1$
logit(γ_{1jk}) = -2.239(0.115)cons.(<=F)$_{ijk}$ + -1.948(0.102)gcseavnormal.(<=F)$_{ijk}$ + h_{jk}
logit(γ_{2jk}) = -1.060(0.093)cons.(<=E)$_{ijk}$ + -1.947(0.090)gcseavnormal.(<=E)$_{ijk}$ + h_{jk}
logit(γ_{3jk}) = -0.000(0.086)cons.(<=D)$_{ijk}$ + -1.927(0.086)gcseavnormal.(<=D)$_{ijk}$ + h_{jk}
logit(γ_{4jk}) = 1.236(0.093)cons.(<=C)$_{ijk}$ + -2.018(0.092)gcseavnormal.(<=C)$_{ijk}$ + h_{jk}
logit(γ_{5jk}) = 3.182(0.143)cons.(<=B)$_{ijk}$ + -2.336(0.123)gcseavnormal.(<=B)$_{ijk}$ + h_{jk}
$h_{jk} = v_{5k}$cons.12345
$[v_{5k}]$ ~ N(0, Ω_v) : Ω_v = [0.677(0.114)]
cov(y_{sjk},y_{tjk}) =γ_{sjk}(1 - γ_{tjk})/cons$_{jk}$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

The relatively small differences among the five coefficients for the GCSE variables, particularly among the first four, suggest that the simpler model with a common coefficient is reasonable. We now return to that model and add a quadratic GCSE term and a gender term.

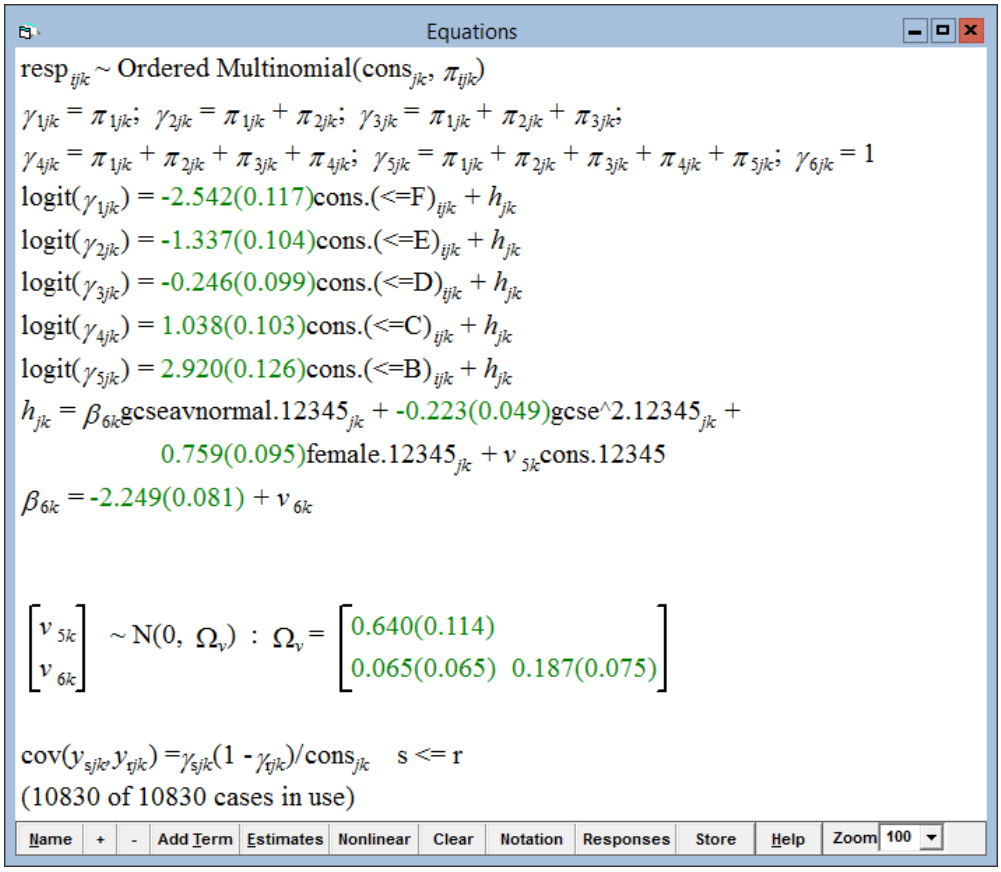
- Click on each of the five separate **gcseavnormal.(<=*)** terms in turn and delete it
- Add **gcseavnormal** (using the **Add Term** button), and choose **add Common coefficient**
- Add **gcse^2** and **gender** using the same process

When we fit this model we obtain the following estimates:


Equations
resp$_{ijk}$ ~ Ordered Multinomial(cons$_{jk}$, π_{ijk})
$\gamma_{1jk} = \pi_{1jk}$; $\gamma_{2jk} = \pi_{1jk} + \pi_{2jk}$; $\gamma_{3jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk}$;
$\gamma_{4jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk}$; $\gamma_{5jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk} + \pi_{5jk}$; $\gamma_{6jk} = 1$
logit(γ_{1jk}) = -2.497(0.115)cons.(<=F)$_{ijk}$ + h_{jk}
logit(γ_{2jk}) = -1.319(0.103)cons.(<=E)$_{ijk}$ + h_{jk}
logit(γ_{3jk}) = -0.246(0.098)cons.(<=D)$_{ijk}$ + h_{jk}
logit(γ_{4jk}) = 1.020(0.102)cons.(<=C)$_{ijk}$ + h_{jk}
logit(γ_{5jk}) = 2.858(0.124)cons.(<=B)$_{ijk}$ + h_{jk}
h_{jk} = -2.189(0.069)gcseavnormal.12345$_{jk}$ + -0.234(0.045)gcse^2.12345$_{jk}$ +
0.753(0.094)female.12345$_{jk}$ + v_{5k}cons.12345
$\begin{bmatrix} v_{5k} \end{bmatrix}$ ~ N(0, Ω_v) : Ω_v = $\begin{bmatrix} 0.642(0.109) \end{bmatrix}$
cov(y_{sjk},y_{tjk}) = γ_{sjk}(1 - γ_{tjk})/cons$_{jk}$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

We can compare the results of fitting this model with our findings from section 11.2. We see that we would make similar inferences about the common effect of gender and GCSE score, but now we have a more detailed description of the probabilities of obtaining each grade.

We will now let the coefficient for the normalised GCSE score be random. We obtain:

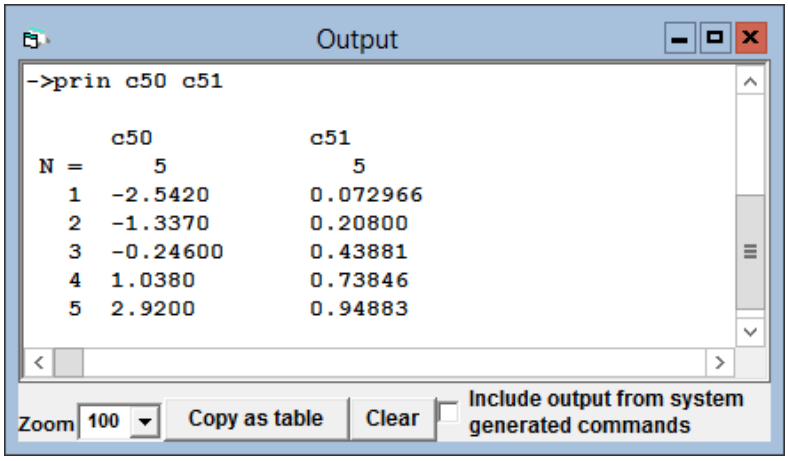

Equations
resp$_{ijk}$ ~ Ordered Multinomial(cons$_{jk}$, π_{ijk})
$\gamma_{1jk} = \pi_{1jk}$; $\gamma_{2jk} = \pi_{1jk} + \pi_{2jk}$; $\gamma_{3jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk}$;
$\gamma_{4jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk}$; $\gamma_{5jk} = \pi_{1jk} + \pi_{2jk} + \pi_{3jk} + \pi_{4jk} + \pi_{5jk}$; $\gamma_{6jk} = 1$
logit(γ_{1jk}) = -2.542(0.117)cons.(<=F)$_{ijk}$ + h_{jk}
logit(γ_{2jk}) = -1.337(0.104)cons.(<=E)$_{ijk}$ + h_{jk}
logit(γ_{3jk}) = -0.246(0.099)cons.(<=D)$_{ijk}$ + h_{jk}
logit(γ_{4jk}) = 1.038(0.103)cons.(<=C)$_{ijk}$ + h_{jk}
logit(γ_{5jk}) = 2.920(0.126)cons.(<=B)$_{ijk}$ + h_{jk}
h_{jk} = β_{6k}gcseavnormal.12345$_{jk}$ + -0.223(0.049)gcse^2.12345$_{jk}$ +
0.759(0.095)female.12345$_{jk}$ + v_{5k}cons.12345
β_{6k} = -2.249(0.081) + v_{6k}
$\begin{bmatrix} v_{5k} \\ v_{6k} \end{bmatrix}$ ~ N(0, Ω_v) : Ω_v = $\begin{bmatrix} 0.640(0.114) & \\ 0.065(0.065) & 0.187(0.075) \end{bmatrix}$
cov(y_{sjk},y_{tjk}) = γ_{sjk}(1 - γ_{tjk})/cons$_{jk}$ s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

To interpret this model, we first consider the fixed part. For boys (**gender** = 0) with average GCSE values (**gcseavnormal** = **gcse**^{2} = 0), we can derive the predicted values of the cumulative category proportions using (11.5). To do this, find the antilogits of the above intercept coefficient estimates by entering the following commands:

- In the **Command interface** window, enter the commands:

```
► join -2.542 -1.337 -0.246 1.038 2.92 c50
► calc c51 = alog(c50)
► print c50 c51
```

The output window will show the following:



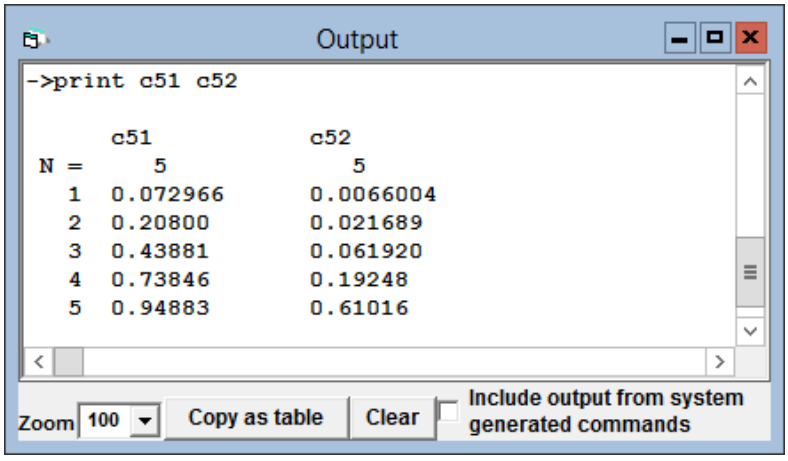
Column **c51** now contains the cumulative probabilities for boys with average GCSE scores: 0.073 for grade F or less, 0.208 for grade E or less and so on. We then difference the probabilities as in (11.5) to obtain the predicted category probabilities (from F to A): (0.072966 0.13504 0.23080 0.29966 0.21036 0.051174).

We might want to see how these figures change for other patterns of explanatory variables, in the case, for example, of boys with an average GCSE score of +1 standard deviation. To do this, enter the following commands:

- Enter the commands:

```
► calc c52 = alog(c50 - 2.249 - 0.223)
► print c51 c52
```

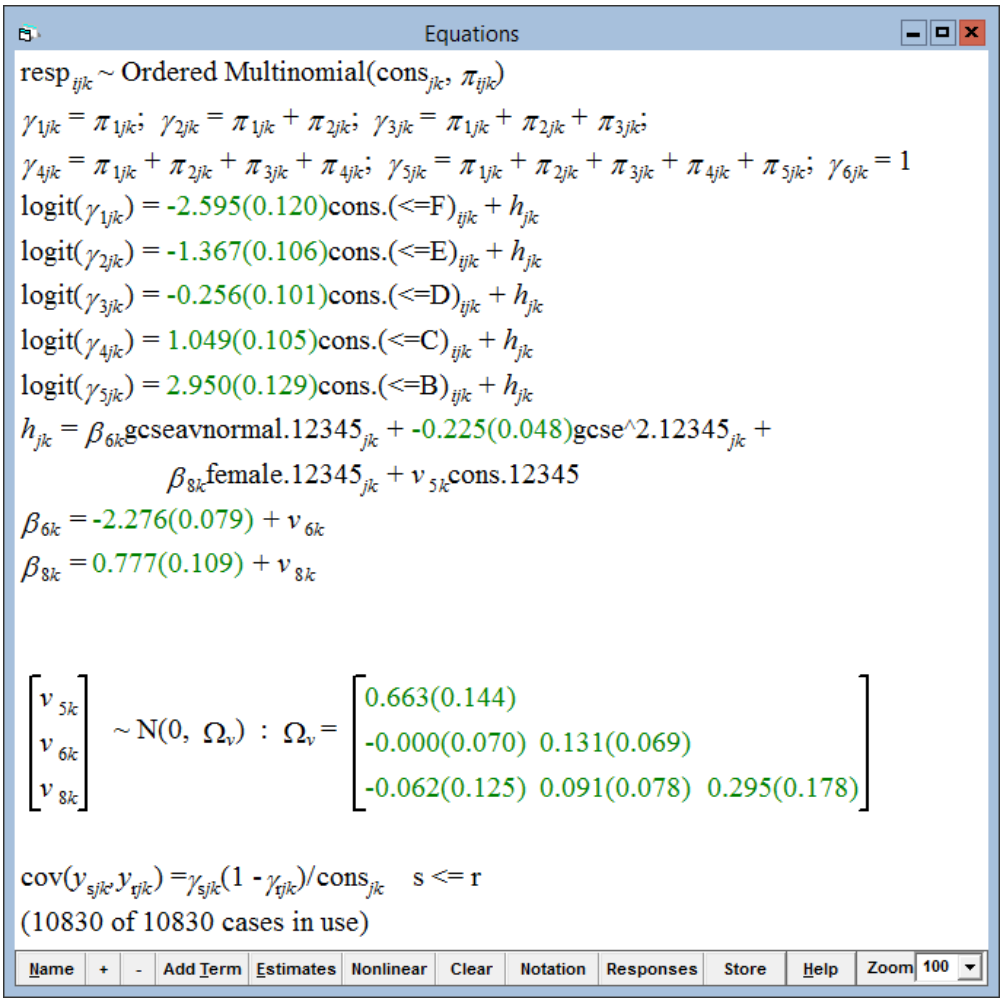
We get:



We can see that, for boys, an increase of 1 SD from the mean, on the GCSE score, has dramatic effects on the cumulative probabilities.

We can also interpret antilogits of the coefficients in the cumulative logit model in terms of odds ratios as in ordinary logit models. Thus for boys at the average GCSE score, the odds of being in grades F or E are $0.21/(1-0.21) = 0.27$.

Finally, we allow the coefficient of gender to vary at the school level and obtain the following result:


Equations
resp_ijk ~ Ordered Multinomial(cons_jk, π_ijk)
γ1jk = π1jk; γ2jk = π1jk + π2jk; γ3jk = π1jk + π2jk + π3jk;
γ4jk = π1jk + π2jk + π3jk + π4jk; γ5jk = π1jk + π2jk + π3jk + π4jk + π5jk; γ6jk = 1
logit(γ1jk) = -2.595(0.120)cons.(<=F)ijk + hjk
logit(γ2jk) = -1.367(0.106)cons.(<=E)ijk + hjk
logit(γ3jk) = -0.256(0.101)cons.(<=D)ijk + hjk
logit(γ4jk) = 1.049(0.105)cons.(<=C)ijk + hjk
logit(γ5jk) = 2.950(0.129)cons.(<=B)ijk + hjk
hjk = β6kgcseavnormal.12345jk + -0.225(0.048)gcse^2.12345jk +
β8kfemale.12345jk + v5kcons.12345
β6k = -2.276(0.079) + v6k
β8k = 0.777(0.109) + v8k
v5k, v6k, v8k ~ N(0, Ωv) : Ωv = 0.663(0.144)
-0.000(0.070) 0.131(0.069)
-0.062(0.125) 0.091(0.078) 0.295(0.178)
cov(ysjk, yrjk) = γsjk(1 - γrjk)/consjk s <= r
(10830 of 10830 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

This suggests that while girls, overall, make less progress in Chemistry between GCSE and A level, this does vary across schools. The estimated between-school standard deviation of this effect is $\sqrt{0.296} = 0.54$, which is only slightly less than the average effect of gender. This suggests that in some schools the girls actually make more progress.

Chapter learning outcomes

- ⋆ How to formulate a cumulative proportional odds model

- ⋆ How to set up and fit such a model in MLwiN
- ⋆ How to interpret the results of such a model

Chapter 12

Modelling Count Data

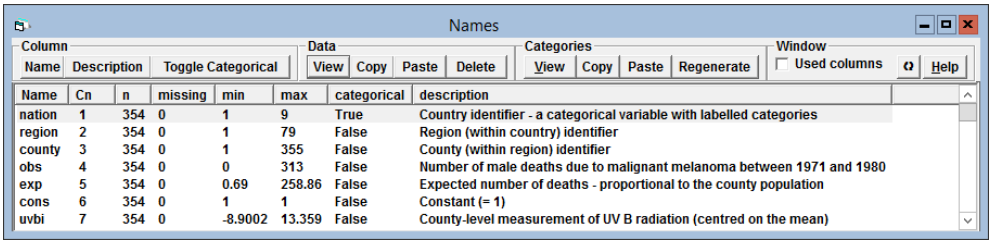
12.1 Introduction

In health and social research it is quite common for the response variable of interest to consist of counts of individuals in a particular state or events of a particular type. For example, we may be interested in the number of children in each of a set of health regions who are hospitalised in a particular year for asthma. For small geographic areas with tiny populations, we could use the binomial distribution in modelling our counts. Usually, however, for large areas with hundreds or thousands of individuals at risk, we would choose the Poisson distribution in our modelling, especially when the number of occurrences of interest in each region is relatively small.

The example data set: malignant melanoma mortality in the European Community

The example we will use in this chapter to illustrate the fitting of multilevel Poisson models for count data comes from the field of environmental health. The problem involves assessing the effect of UV radiation exposure on the mortality rate due to malignant melanoma in the European Community. Further information about the study is reported in Langford et al. (1998)

Open the data set **mmmec.ws**. The **Names** window shows:



Name	Cn	n	missing	min	max	categorical	description
nation	1	354	0	1	9	True	Country identifier - a categorical variable with labelled categories
region	2	354	0	1	79	False	Region (within country) identifier
county	3	354	0	1	355	False	County (within region) identifier
obs	4	354	0	0	313	False	Number of male deaths due to malignant melanoma between 1971 and 1980
exp	5	354	0	0.69	258.86	False	Expected number of deaths - proportional to the county population
cons	6	354	0	1	1	False	Constant (= 1)
uvbi	7	354	0	-8.9002	13.359	False	County-level measurement of UV B radiation (centred on the mean)

The variables are defined as follows:

Variable	*Description*
nation	Country identifier – a categorical variable with labelled categories
region	Region (within country) identifier
county	County (within region) identifier
obs	Number of male deaths due to malignant melanoma between 1971 and 1980
exp	Expected number of deaths – proportional to the county population
cons	Constant (= 1)
uvbi	County-level measurement of UV B radiation (centred on the mean)

12.2 Fitting a simple Poisson model

Count data are constrained to be non-negative. If we were to try fitting a Normal model to the data, we could produce predicted counts that were negative, so we would prefer to model the logarithms of the counts. We will therefore fit a Poisson model to the count data using a log link function.

We are actually more interested in the rates of malignant melanoma mortality rather than the actual counts, as each geographic unit will have a different population size. If we were to use the raw counts of deaths, the units with larger population size would have larger counts thus masking the true relationships with explanatory variables. To work with the rates rather than the counts, we use an additional parameter known as an *offset*.

This offset is set to be equal to the log (base e) of the expected death count (which is based on county population). If y_i is the observed count in county i, π_i is the mean of the Poisson distribution for the county and E_i is the expected count or offset, we can express a single-level Poisson model as follows:

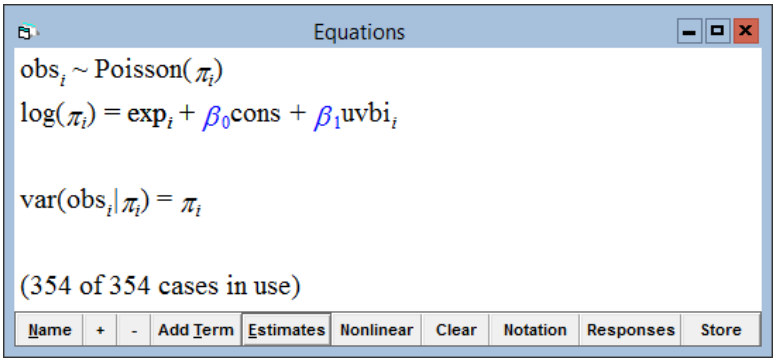
$$\begin{aligned} y_i &\sim \text{Poisson}(\pi_i) \\ \log(\pi_i) &= \log(E_i) + X\beta \\ \log(\pi/E_i) &= X\beta \quad \text{(alternative formulation)} \end{aligned} \tag{12.1}$$

To specify this model in MLwiN:

- Open the **Equations** window and click on y
- In the Y variable window, choose **obs** from the **y:** drop down list
- Select **N levels:** as **1 - i**

- Select **county** from the **level 1(i)** drop-down list and click **Done**
- Click on the **N** that appears on the first line on the **Equations** window
- Set distribution type to **Poisson** and click **Done**
- Use the **Add term** button to add **cons** and **uvbi** to the model
- Click on π_i in the second equation in the **Equations** window
- In the **specify offset** window, select **exp** from the drop down list and click **Done**
- Click on the **Estimates** button in the **Equations** window

The **Equations** window will now look like this:



Note that the final line in the window reflects the fact that the variance of a Poisson variable with mean π is also π. We have included ***exp*** *as an offset, but from equation* (12.1) *we see we need to use log(****exp****) instead.*

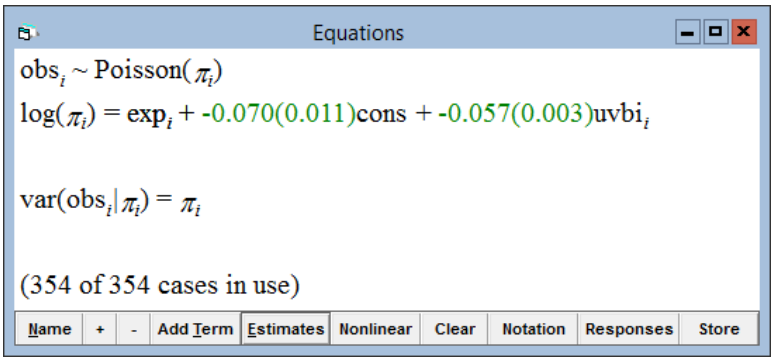
- In the **Command interface** window, type the command

```
► calc 'exp' = loge('exp')
```

Set the estimation procedure:

- Click on the **Nonlinear** button in the **Equations** window
- In the **Nonlinear Estimation** window, select **Poisson**, **1st order** and **MQL**
- Click **Done**
- Click **Start**

You should obtain the following results:



We can see here that in this model there is a negative relationship between incidence of melanoma and UV exposure. This seems surprising but may be explained by including more structure into the data.

12.3 A three-level analysis

We now consider a three-level Poisson model that will allow us to examine geographic variation in melanoma mortality. Begin by setting up the hierarchical structure in MLwiN:

- In the **Equations** window, click on **obs**
- In the Y variable window, set **N levels:** to be **3 - ijk**
- Set level 2 as **region** and level 3 as **nation,** and click **done**

The first model we wish to consider is a simple variance components model to examine the nation and region effects on mortality without adjusting for UV exposure. To do this:

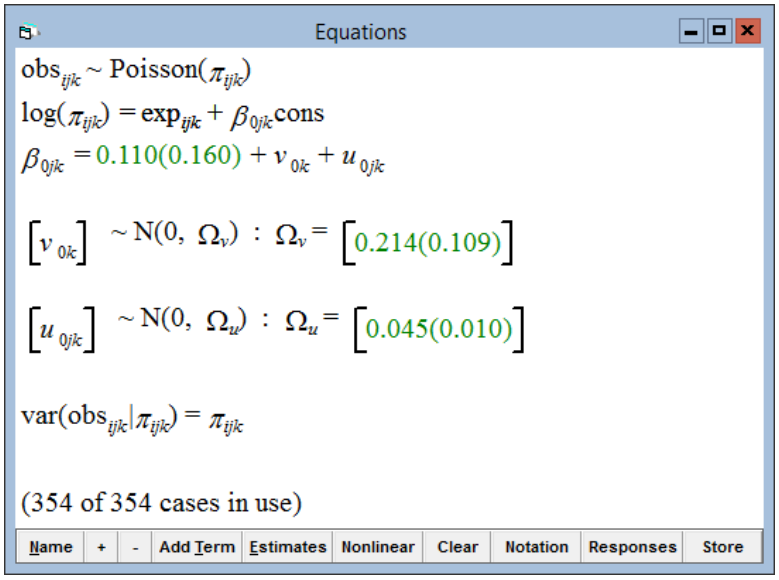
- Remove **uvbi** from the model
- Set **cons** to be random at both the **nation** and **region** levels

We will assume in this model that random error terms at the two levels are Normally distributed.

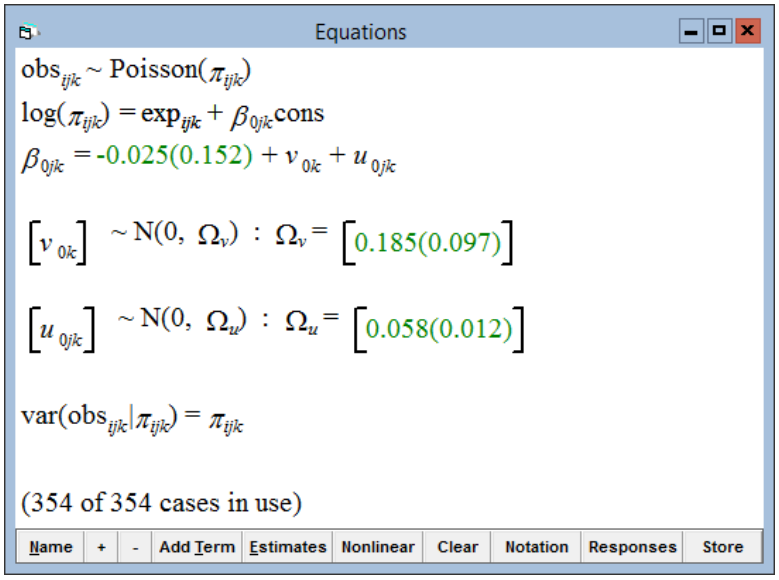
Given we have only nine countries it is advisable to use RIGLS estimation, which provides less biased estimates of the variance than IGLS when the number of highest-level units is small.

- Select **RIGLS** from the **Estimation** menu and run the model

The results are as follows:


Equations
obs$_{ijk}$ ~ Poisson(π_{ijk})
log(π_{ijk}) = exp$_{ijk}$ + β_{0jk}cons
β_{0jk} = 0.110(0.160) + v_{0k} + u_{0jk}
[v_{0k}] ~ N(0, Ω_v) : Ω_v = [0.214(0.109)]
[u_{0jk}] ~ N(0, Ω_u) : Ω_u = [0.045(0.010)]
var(obs$_{ijk}$|π_{ijk}) = π_{ijk}
(354 of 354 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

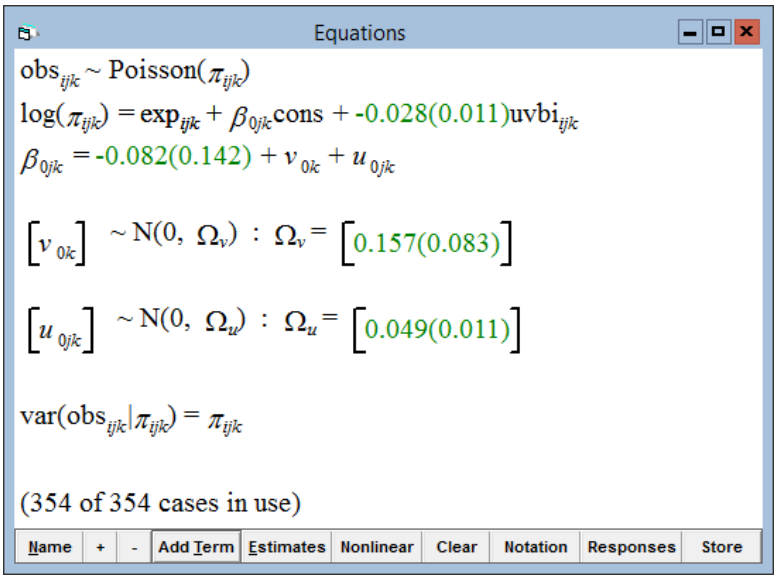
This model shows that there is almost five times as much variability between nations as there is between regions within nations. The 1st order MQL method is not as accurate as the 2nd order PQL method, so we will now fit the same model with the latter method (accessed via the Nonlinear button). The results are as follows:


Equations
obs$_{ijk}$ ~ Poisson(π_{ijk})
log(π_{ijk}) = exp$_{ijk}$ + β_{0jk}cons
β_{0jk} = -0.025(0.152) + v_{0k} + u_{0jk}
[v_{0k}] ~ N(0, Ω_v) : Ω_v = [0.185(0.097)]
[u_{0jk}] ~ N(0, Ω_u) : Ω_u = [0.058(0.012)]
var(obs$_{ijk}$|π_{ijk}) = π_{ijk}
(354 of 354 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The results for 2nd order PQL are similar to the MQL results but with a different split of the variance between the two levels. One important difference with Poisson models that has also been shown by some simulations is that the MQL method tends to overestimate some of the variance parameters. This is in contrast with other types of discrete response model where

this method tends to underestimate the variance. For the remainder of the models in this chapter we will only use the 2nd order PQL method.

The next step in our model fitting with this data set is to add the predictor uvbi into the multilevel model. The results are as follows:


Equations
obs$_{ijk}$ ~ Poisson(π_{ijk})
log(π_{ijk}) = exp$_{ijk}$ + β_{0jk}cons + -0.028(0.011)uvbi$_{ijk}$
β_{0jk} = -0.082(0.142) + v_{0k} + u_{0jk}
[v_{0k}] ~ N(0, Ω_v) : Ω_v = [0.157(0.083)]
[u_{0jk}] ~ N(0, Ω_u) : Ω_u = [0.049(0.011)]
var(obs$_{ijk}$|π_{ijk}) = π_{ijk}
(354 of 354 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

This model again shows that the variability between nations is three times the variability between the regions within nations. The amount of UV radiation still has a significant negative effect on melanoma mortality.

12.4 A two-level model using separate country terms

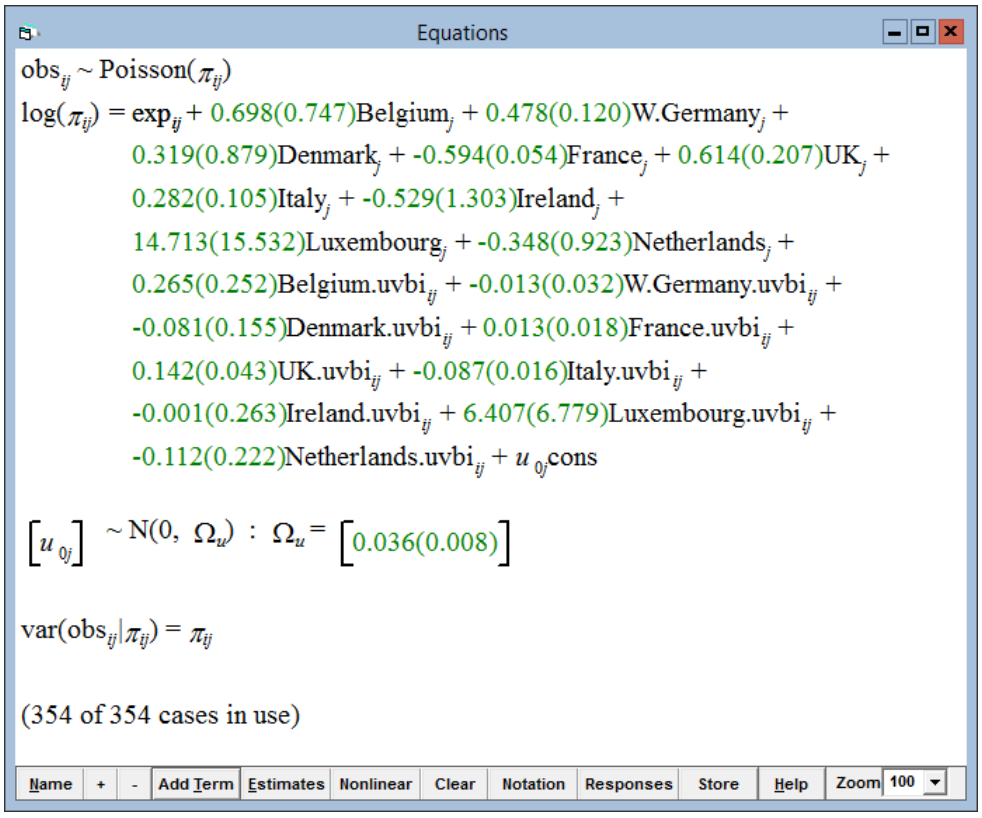
In our three-level model we have very few level 3 units, so the accuracy of the level 3 variance estimate is low. An alternative approach that we can use in such a situation would involve fitting a two-level model with a separate (fixed) intercept for each level 3 unit. We do this here by including country indicators in (just) the fixed part of the model.

Let's now set up a model with a separate term for each country. We will also allow separate **uvbi** slopes for each country to see if this sheds any light on the counterintuitive finding that increased **uvbi** exposure decreases melanoma counts.

- Click on **obs**$_{ijk}$ and in the **Y variable** window, set **N levels:** to **2-ij** and click **done**
- Remove **uvbi** from the model

- Click on **cons** and in the **X variable** window, uncheck the **Fixed Parameter** check box
- Click **Done**
- Click on **Add Term**
- In the **Specify term** window, set **variable** to **nation**, set **reference category** to [**None**] and click **Done**
- Click on **Add Term** again
- In the **Specify term** window, set **order** to **1**
- Set the first variable to **nation** and set the second variable to **uvbi**
- Set the **reference category** for **nation** to [**None**] and click **Done**

Running this model produces:

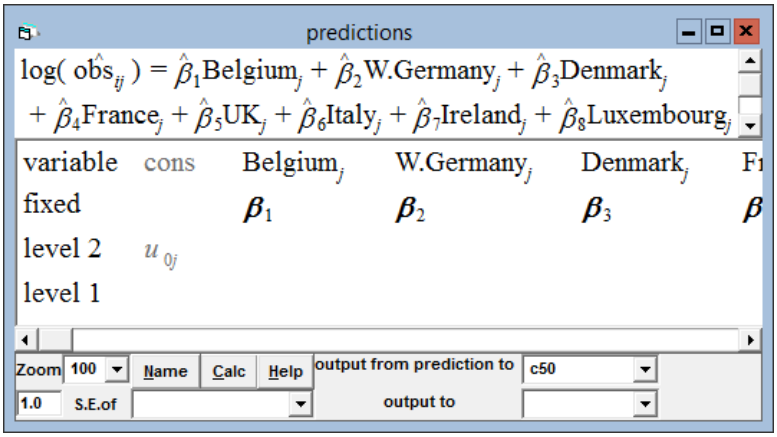

Equations
obs_ij ~ Poisson(π_ij)
log(π_ij) = exp_ij + 0.698(0.747)Belgium_j + 0.478(0.120)W.Germany_j +
0.319(0.879)Denmark_j + -0.594(0.054)France_j + 0.614(0.207)UK_j +
0.282(0.105)Italy_j + -0.529(1.303)Ireland_j +
14.713(15.532)Luxembourg_j + -0.348(0.923)Netherlands_j +
0.265(0.252)Belgium.uvbi_ij + -0.013(0.032)W.Germany.uvbi_ij +
-0.081(0.155)Denmark.uvbi_ij + 0.013(0.018)France.uvbi_ij +
0.142(0.043)UK.uvbi_ij + -0.087(0.016)Italy.uvbi_ij +
-0.001(0.263)Ireland.uvbi_ij + 6.407(6.779)Luxembourg.uvbi_ij +
-0.112(0.222)Netherlands.uvbi_ij + u_0jcons
[u_0j] ~ N(0, Ω_u) : Ω_u = [0.036(0.008)]
var(obs_ij|π_ij) = π_ij
(354 of 354 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

We can now form predicted lines for each of the nine countries.

- Select the **Predictions** window from the **Model** menu
- Click on **fixed** in the lower panel and select **Include all fixed coefficients** from the resulting list
- Beside **output from prediction to**, select **c50**

The **Predictions** window should now be as follows (we have not shown all

of it):


predictions
log(obs$_{ij}$) = $\hat{\beta}_1$Belgium$_j$ + $\hat{\beta}_2$W.Germany$_j$ + $\hat{\beta}_3$Denmark$_j$
+ $\hat{\beta}_4$France$_j$ + $\hat{\beta}_5$UK$_j$ + $\hat{\beta}_6$Italy$_j$ + $\hat{\beta}_7$Ireland$_j$ + $\hat{\beta}_8$Luxembourg$_j$
variable cons Belgium$_j$ W.Germany$_j$ Denmark$_j$ F1
fixed β_1 β_2 β_3 β
level 2 u_{0j}
level 1
Zoom 100 Name Calc Help output from prediction to c50
1.0 S.E.of output to

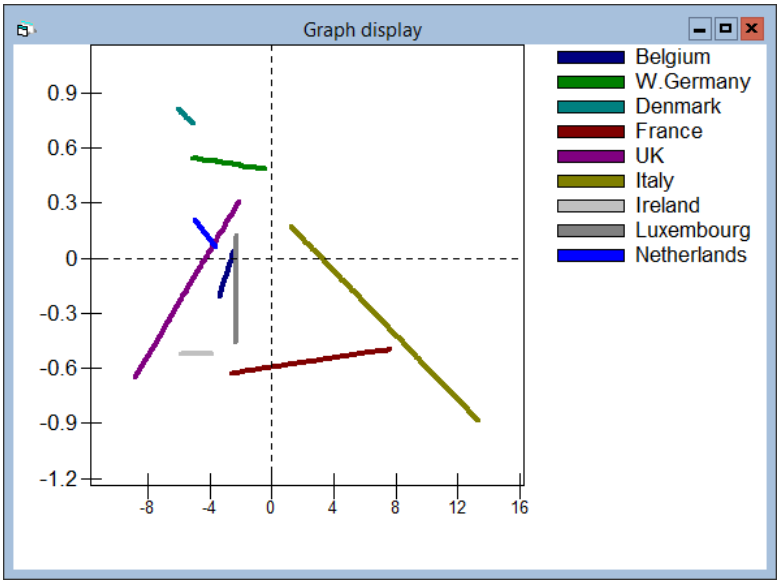
Complete the estimation of the predicted values:

- Click on **Calc** to store the predicted values in **c50**

We would now like to plot out nine lines, one for each country. To do this we need to open the **Customised graph** window from the **Graphs** menu.

- In the **Customised graph** window set **y** to be **c50** and **x** to be **uvbi**
- Select **group** as **nation**
- Select **plot type** as **line**
- On the **plot style** tab select **colour** as **16 rotate** and **line thickness** as **4**
- Click the **other** tab, and beneath the **construct key labels from** area, check **group code** to construct a key
- Click **Apply**.

Selecting all these options should produce the following graph:



On your computer screen you will be able to identify the lines according to the colour coding. Remember that the predicted values are the logarithm to the base e of the relative risk. Here effects of the UV radiation appear to vary dramatically from nation to nation. From the estimates in the **Equations** window we saw that all countries except country 5 (UK) and country 6 (Italy) have estimated effects that are not significantly different from zero. This graph shows a clearer picture of the actual effects of **uvbi** in each country as we can clearly see that there is very little overlap in terms of uvbi values between some countries.

Luxembourg (country 8) is poorly estimated because it contains only three counties. We can see that even though the intercept term for Luxembourg is huge, in fact there are no people in Luxembourg who experience the mean UV exposure (that the intercept represents). For the values of UV exposure experienced in Luxembourg, the relative risk is close to zero.

The UK has a strong positive association between UV radiation and melanoma mortality, and this could be explained in many ways. One reason could be the combination of few hot sunny days at home combined with more recreational travel to warmer climates. Italy on the other hand, has a negative association between UV radiation and melanoma mortality. This could possibly be explained by a higher prevalence of low risk dark skinned people in the south of Italy, which has a higher UV exposure.

This model ends our examination of the melanoma mortality data set. To finish the chapter we will consider some general issues involving discrete response models.

12.5 Some issues and problems for discrete response models

The binomial response models of the previous chapter and the count data response models of the present chapter are both examples of multilevel *Generalised linear models* (McCullagh & Nelder, 1989).

In addition to fitting a Poisson error model for count data we can also fit a negative binomial error that allows greater flexibility, essentially allowing a more complex variance structure than that associated with the Poisson distribution. The negative binomial appears as an extra option on the drop down menu list for error distribution. See Goldstein (2003), Chapter 7.

There are several problems in fitting such generalised linear models, and some of these have been touched upon in this chapter. Research is being actively carried out in this area and the development of future software has been planned to reflect this. It will be shown how to fit Generalised linear models using MCMC methods and bootstrapping in later chapters. In general it is recommended that more than one approach be tried, and if similar results are obtained from the various estimation methods then the analyst can have some confidence in the estimates.

Some care is also needed with using any of the standard diagnostic procedures based on estimated residuals in binary response models. Where there are few level 1 units per level 2 unit and / or the underlying probabilities are close to 0 or 1, then these estimates are not approximately Normally distributed, even when the model is correct.

Chapter learning outcomes

- ⋆ How to fit Poisson models to count data
- ⋆ How to use an offset in MLwiN to model rates rather than raw counts
- ⋆ How to fit single level models in MLwiN
- ⋆ How to define categorical variables in MLwiN
- ⋆ How to use dummy variables to reduce the number of levels in a model when there are few higher-level units

Chapter 13

Fitting Models to Repeated Measures Data

13.1 Introduction

Repeated measures data arise in a number of contexts, such as child or animal growth, panel surveys and the like. The basic structure is that of measurements nested within subjects, i.e. a two-level hierarchy.

Suppose, for example, we have a sample of students whose reading attainment is measured on a number of occasions. The students define level two, and the repeated measures or occasions define level one. In longitudinal repeated measures designs, we usually have a large number of level two units with rather few level one units in each, in contrast to the cross-sectional study that provided the data we analysed in Chapter 2.

We can, of course, extend this structure to include a third level representing groups of students such as classes or schools. It is also worth bearing in mind that our repeated measures could be obtained from schools or teachers rather than (or even as well as) from students. So we might have a four-level structure, with a sample of schools, studied over time by measuring successive cohorts of students, and these students themselves repeatedly measured as they pass through the school. A study with such a design would clearly be large and complex, but it would have the potential for assessing the stability of school effects as well as for studying students' educational growth.

In the following sections we introduce a data set from a longitudinal study of student achievement, and formulate and analyse a sequence of models of increasing complexity. We shall, however, only cover some of the possible elaborations of the basic models. Repeated measures models can be extended to the case of complex serial correlation structures at level 1, and to the multivariate case. We shall also not deal with the case of repeated discrete

(for example, binary) responses since this raises some new issues that are currently being investigated.

Statistical models for repeated measures data

In multilevel structures we do not require balanced data to obtain efficient estimates. In other words, it is not necessary to have the same number of lower-level units within each higher-level unit. With repeated measures data we do not require the same number of measurement occasions per individual subject (level 2). Often in longitudinal studies individuals leave the study or miss one or more measurement occasions. Nevertheless, all of the available data can be incorporated into the analysis. This assumes that the probability of being missing is independent of any of the random variables in the model. This condition, known as *completely random dropout* (CRD) may be relaxed to that of *random dropout* (RD) where the missing mechanism depends on the observed measurements. In this latter case, so long as a full information estimation procedure is used, such as that of maximum likelihood in MLwiN for Normal data, then the actual missingness mechanism can be ignored. See Diggle & Kenward (1994) for a discussion of this issue. The ability to handle unbalanced data is in contrast to analyses based upon 'repeated measures analysis of variance' (See Plewis (1997), Chapter 4).

We can adopt two perspectives on repeated measures data. The first is concerned with what we may term 'growth curves' since such models were originally developed to fit human and animal anthropometric data (Goldstein, 1979). Some examples are illustrated in Figure 13.1.

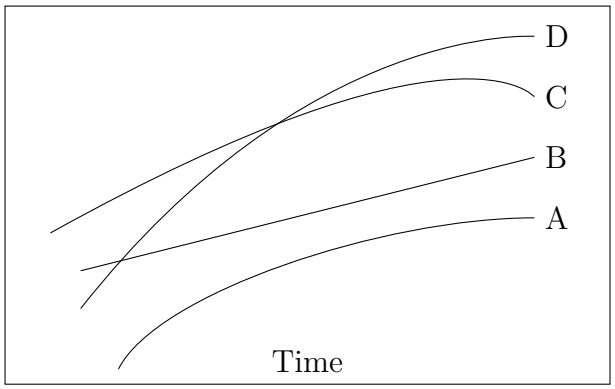


Figure 13.1: Some examples of growth curves

The curves show different kinds of change in a variable with time or age. Growth is linear over the age range for case B, non-linear but monotonic for cases D and A and non-linear and non-monotonic for case C.

A second way of looking at repeated measures data, when there is a small number of fixed occasions, is to use a 'conditional' model where later mea-

sures are related to one or more earlier measures. This can be a useful approach in some circumstances, e.g., in a designed experiment in which each subject is measured before receiving an intervention, immediately afterward and on a follow-up occasion. This approach raises no new multilevel modelling issues. In this chapter we are only concerned with the first type of modelling, i.e., for repeated measures growth curve data. We now look at an example data set and analysis.

Repeated measures data on reading attainment

The data we are going to use come from a longitudinal study of a cohort of students who entered 33 multi-ethnic inner London infant schools in 1982, and who were followed until the end of their junior schooling in 1989. More details about the study can be found in Tizard et al. (1988). Students' reading attainments were tested on up to six occasions; annually from 1982 to 1986 and in 1989. Reading attainment is the response, and there are three levels of data - school (level 3), student (level 2) and measurement occasion (level 1). In addition, there are three explanatory variables. The first is the student's age, which varies from occasion to occasion and is therefore a level one variable. The other two are gender (coded 0 for males and 1 for females) and ethnic group (coded 0 for white and 1 for black). These vary from student to student and are thus level two variables. The initial sample at school entry consisted of 171 white indigenous students and 106 black British students of African Caribbean origin. The sample size increased to 371 one year later and fell to 198 by the end of junior school.

Some basic questions we could investigate are:

1. How does reading attainment change as students get older?
2. Does this vary from student to student?
3. Do different subgroups of students, e.g. boys and girls, have different patterns of change?

In this chapter we will explore the first two of these questions.

Table 13.1 below gives the number of reading tests per student and shows that only a minority of students were measured on every occasion. Altogether, 1758 observations were obtained on 407 students, of whom 259 were white and 148 were black. It is important to note that students with, say, a total of three tests did not necessarily all have tests at the same three occasions. Table 13.2 illustrates some of the response patterns across students. This table also indicates that students' ages can differ at fixed measurement occasions. Compare, for example, student one and student four at occasion one. This underlines another advantage of multilevel modelling of repeated measures

which has already been mentioned, namely the ability to handle unequal measurement intervals.

Table 13.1: Summary of reading test data

# of Tests	# of Students	% of Total Students	Inverse Cumulative %
1	37	9	100
2	41	10	91
3	42	10	81
4	48	12	71
5	113	28	59
6	126	31	31
TOTAL	407	100	n.a.

Table 13.2: Different patterns of test sequences and ages at testing

STUDENT	OCCASION					
	1	2	3	4	5	6
ONE	4.6	5.7	6.7	7.7	8.7	11.6
TWO	4.8	5.7	6.8	-	-	-
THREE	4.8	5.8	-	7.8	-	-
FOUR	4.9	-	-	-	-	-

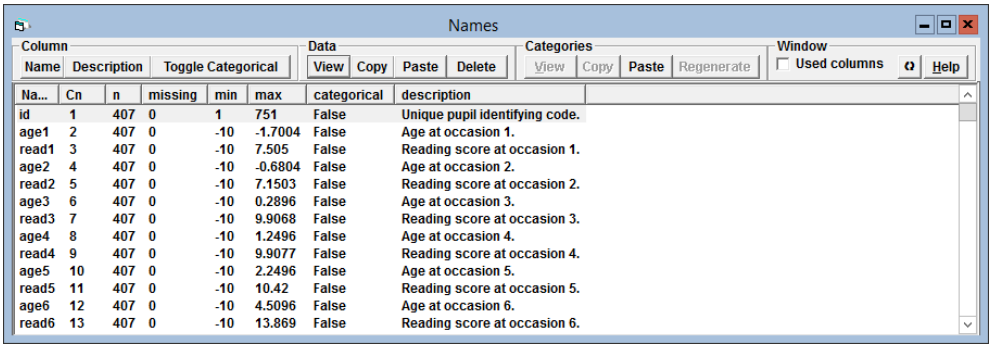
13.2 A basic model

Defining the scale for the response variable

One problem we have with the present data is how to define and construct the response for students at different ages. Reading attainment cannot usually be measured with the same test at each age, and in our example four rather different, age-appropriate tests were used. The underlying construct is reading but the observed variable changes with age, and we wish to construct a common age scale with sensible properties. Moreover, we may find that our results vary as we change the scale of our response. Here we work with a scale for the response defined in the following 'age-equivalent' way. The mean reading score at each occasion is set equal to the mean student age for that occasion, and the variance is set to increase from one occasion to the next in such a way that the coefficient of variation (i.e. the standard deviation divided by the mean) is constant and equal to 0.13. Allowing the variance, as well as the mean, to increase with age is consistent with what we know about many kinds of growth. An alternative measure would be z scores (zero mean and unit variance) at each occasion.

Setting up the data structure

Open the supplied worksheet, **reading1.ws**, which contains 13 variables for 407 students as shown below in the **Names** display:



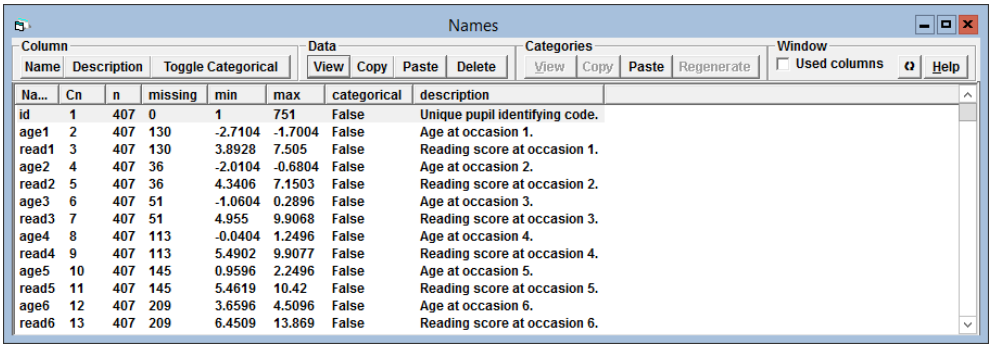
Na...	Cn	n	missing	min	max	categorical	description
id	1	407	0	1	751	False	Unique pupil identifying code.
age1	2	407	0	-10	-1.7004	False	Age at occasion 1.
read1	3	407	0	-10	7.505	False	Reading score at occasion 1.
age2	4	407	0	-10	-0.6804	False	Age at occasion 2.
read2	5	407	0	-10	7.1503	False	Reading score at occasion 2.
age3	6	407	0	-10	0.2896	False	Age at occasion 3.
read3	7	407	0	-10	9.9068	False	Reading score at occasion 3.
age4	8	407	0	-10	1.2496	False	Age at occasion 4.
read4	9	407	0	-10	9.9077	False	Reading score at occasion 4.
age5	10	407	0	-10	2.2496	False	Age at occasion 5.
read5	11	407	0	-10	10.42	False	Reading score at occasion 5.
age6	12	407	0	-10	4.5096	False	Age at occasion 6.
read6	13	407	0	-10	13.869	False	Reading score at occasion 6.

Field or Column number 1 is the student identifier. This is followed by six pairs of fields corresponding to the six occasions, each pair being the student's reading score and age on that occasion.

Note that the ages have been centred on the mean age. In this data set, −10 represents a missing value. We can tell MLwiN that −10 is the missing value code by:

- Select the **Options** menu
- Select **Numbers(Display precision and missing value code)**
- Set the **missing value** code to −10
- Click the **Apply** button, then **Done**

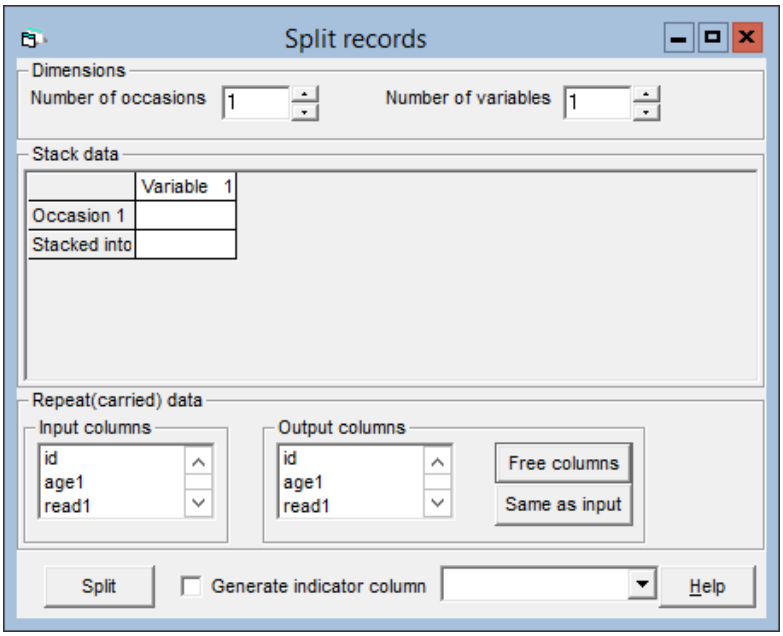
The **Names** window is updated and now explicitly shows the number of missing cases in each variable:



Na...	Cn	n	missing	min	max	categorical	description
id	1	407	0	1	751	False	Unique pupil identifying code.
age1	2	407	130	-2.7104	-1.7004	False	Age at occasion 1.
read1	3	407	130	3.8928	7.505	False	Reading score at occasion 1.
age2	4	407	36	-2.0104	-0.6804	False	Age at occasion 2.
read2	5	407	36	4.3406	7.1503	False	Reading score at occasion 2.
age3	6	407	51	-1.0604	0.2896	False	Age at occasion 3.
read3	7	407	51	4.955	9.9068	False	Reading score at occasion 3.
age4	8	407	113	-0.0404	1.2496	False	Age at occasion 4.
read4	9	407	113	5.4902	9.9077	False	Reading score at occasion 4.
age5	10	407	145	0.9596	2.2496	False	Age at occasion 5.
read5	11	407	145	5.4619	10.42	False	Reading score at occasion 5.
age6	12	407	209	3.6596	4.5096	False	Age at occasion 6.
read6	13	407	209	6.4509	13.869	False	Reading score at occasion 6.

This arrangement of data, in which each row of a rectangular array corresponds to a different individual and contains all the data for that individual, is a natural one, but it does not reflect the hierarchical structure of measurements nested within individuals. The **Split records** window (shown below),

accessed via the **Data Manipulation** menu, is designed to transform an individual's data record into separate records (or rows), one for each occasion. In the present case we shall produce six records per student, that is, 2442 records altogether. The ordering of students will be preserved, and they will become the level 2 units.

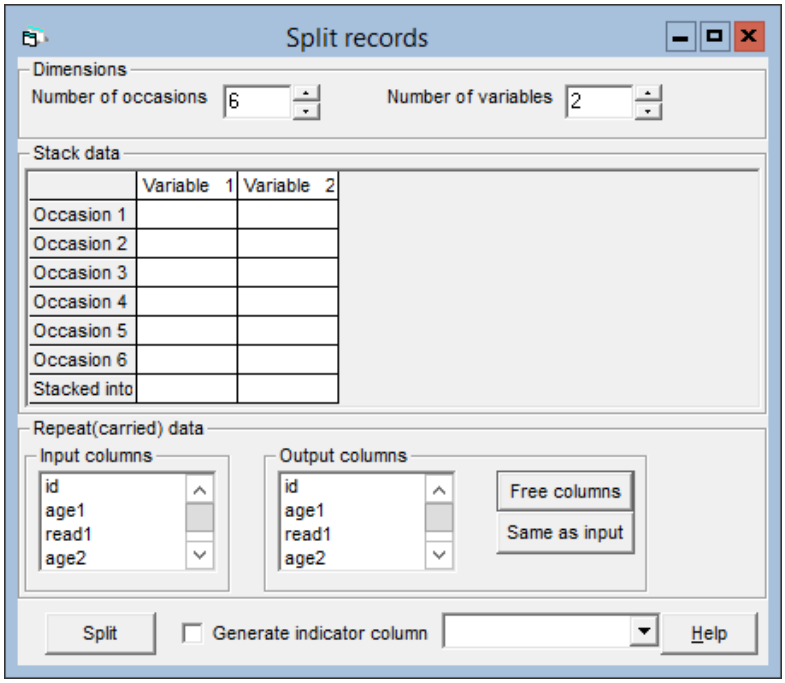


There are two types of data to consider: occasion specific data and repeated data. The former (in principle) change from occasion to occasion, in this case, the reading scores and the ages. The latter remain constant from occasion to occasion, in this case, the student identifiers.

First let us deal with the occasion specific data:

- Open the **Split records** window
- Set the **Number of occasions** to **6**
- Set the **Number of variables** to **2**

Doing this produces:



We need to stack the six reading scores into a single column and the six ages into a single column.

- In the **Stack data** grid, click on **Variable 1**
- From the drop-down list that appears, select the six variables **read1, read2, . . . , read6** and then click **Done**. (To make multiple selections, hold the control key down while clicking on variable names.)
- Repeat the above two steps for **Variable 2** and the six variables **age1** to **age6**

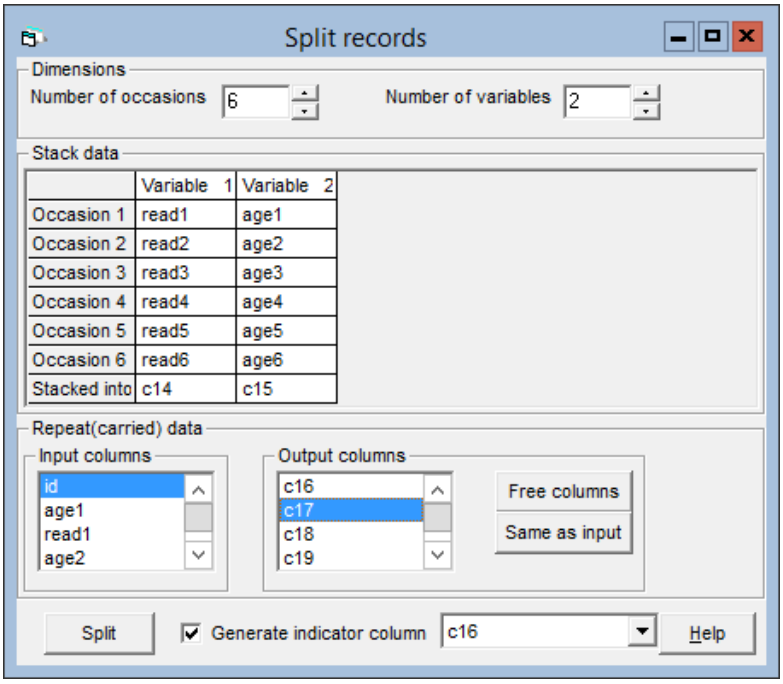
Clicking on the column headings allows you to set all six occasion variables from a single pick list. The first variable on the list is assigned to occasion 1, the second to occasion 2 and so on. This works fine in our case because the variables appear on the list in the correct order. If this is not the case, you can specifically assign variables to occasions by clicking on individual cells in the grid.

- Click in turn on the two empty cells in the **Stacked into** row of the **Stack data** grid
- From the drop-down lists that appear, select **c14** and **c15** respectively.
- Tick the **Generate indicator column** check box
- In the neighbouring drop-down list, select **c16**

That deals with occasion specific data. Now we will specify the repeated data:

- In the **Repeat(carried data)** frame, select **ID** as the input column and **c17** as the output

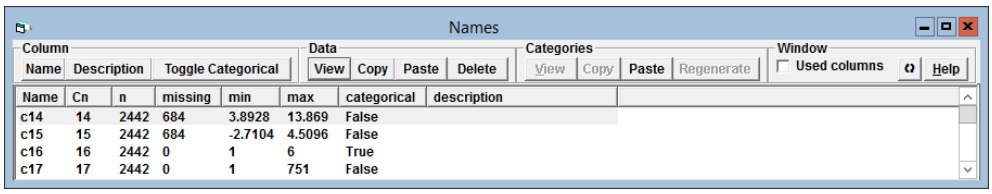
The completed set of entries should look like this:



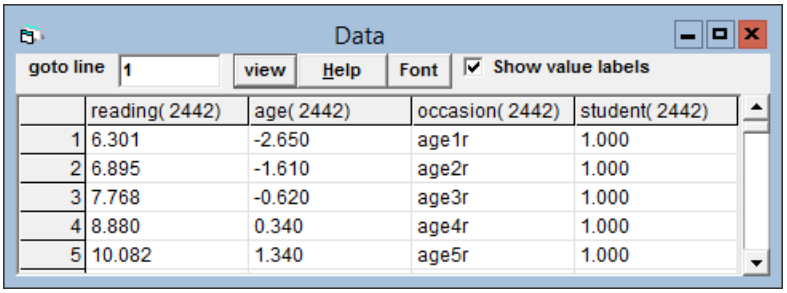
This will take the six reading score variables, each of length 407, and stack them into a single variable in **c14**. The six age variables will be stacked into **c15**. Each **id** code will be repeated six times, and the repeated codes are stored in **c17**. The indicator column, which is output to **c16**, will contain occasion identifiers for the new long data set.

- Click the **Split** button to execute the changes
- You will be asked if you want to save the worksheet — select NO

The **Names** window now shows the following for **c14** through **c17**:



Assign the names **reading, age, occasion** and **student** to **c14-c17.** Viewing columns 14-17 will now show:

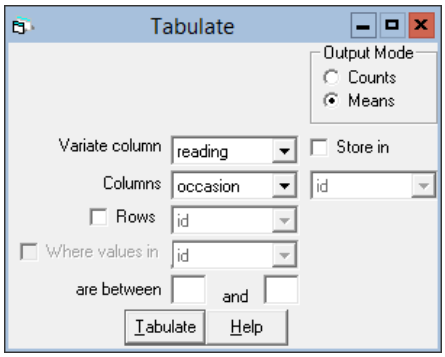


	reading(2442)	age(2442)	occasion(2442)	student(2442)
1	6.301	-2.650	age1r	1.000
2	6.895	-1.610	age2r	1.000
3	7.768	-0.620	age3r	1.000
4	8.880	0.340	age4r	1.000
5	10.082	1.340	age5r	1.000

The data are now in the required form with one row per occasion. It would now be a good idea to save the worksheet, using a different name.

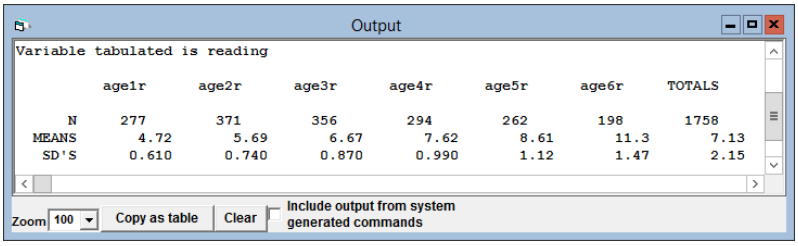
Initial data exploration

Before we start to do any modelling, we can do some exploratory work. The mean reading score at each occasion is obtained using the **Tabulate** window accessed via the **Basic statistics** menu:



- Select **Means** as the **Output Mode**
- A drop-down list labelled **variate column** appears. Select **reading**.
- From the **Columns** drop-down list, select **occasion**
- Click **Tabulate**

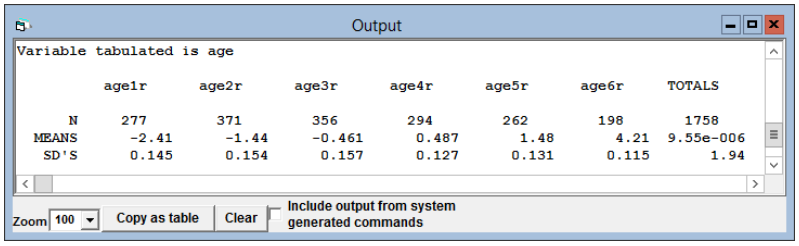
This produces the output :


Output
Variable tabulated is reading
age1r age2r age3r age4r age5r age6r TOTALS
N 277 371 356 294 262 198 1758
MEANS 4.72 5.69 6.67 7.62 8.61 11.3 7.13
SD'S 0.610 0.740 0.870 0.990 1.12 1.47 2.15
Zoom 100
Copy as table
Clear
Include output from system generated commands

As we have noted earlier, our measure of reading is constructed from a series of different reading tests. The present scaling choice is reflected in the models

which follow, where the increasing variance with age is modelled by fitting random coefficients.

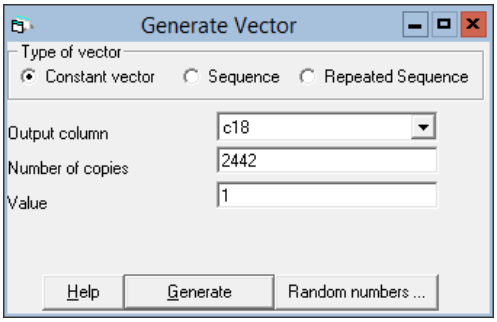
Now use the **Tabulate** window to tabulate mean age by occasion.


Output
Variable tabulated is age
age1r age2r age3r age4r age5r age6r TOTALS
N 277 371 356 294 262 198 1758
MEANS -2.41 -1.44 -0.461 0.487 1.48 4.21 9.55e-006
SD'S 0.145 0.154 0.157 0.127 0.131 0.115 1.94
Zoom 100
Copy as table
Clear
Include output from system generated commands

The age variable has been transformed by measuring it as a deviation from the overall mean age. The mean reading score at each occasion is, from the way we have defined our reading scale, equal to the mean true age at that occasion, not the mean on the transformed age scale.

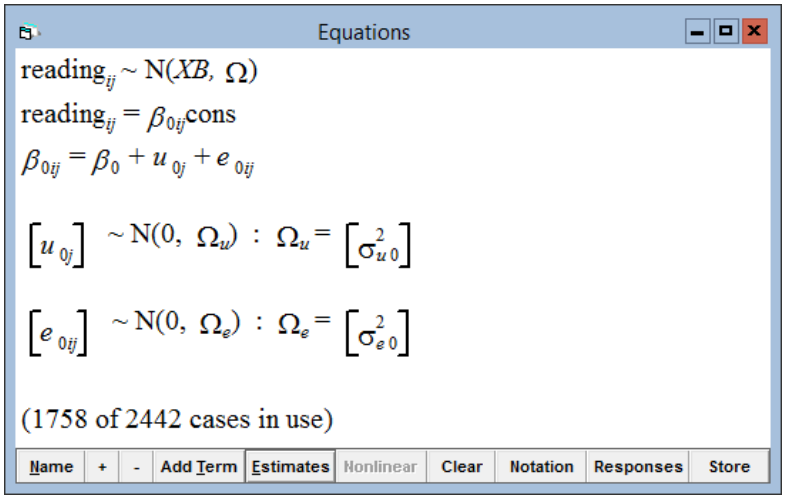
We are now almost in a position to set up a simple model, but first we must define a constant column.

- Access the **Generate Vector** window via the **Data Manipulation** menu
- Fill out the options as shown below and click **Generate**
- Use the **Names** window to assign the name **cons** to **c18**

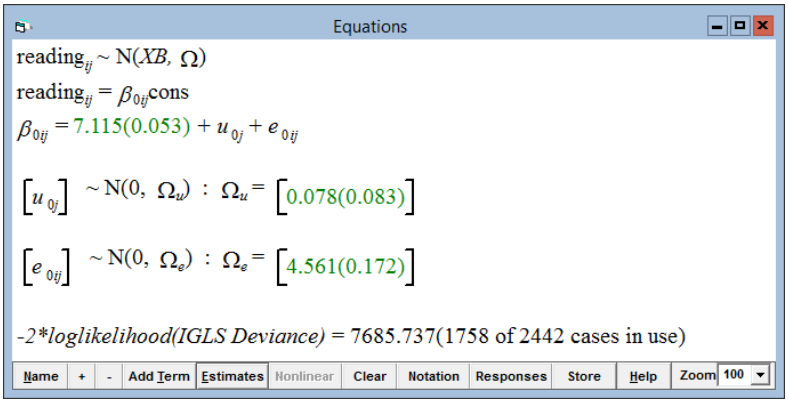


A baseline variance components model

We start by seeing how the total variance is partitioned into two components: between students and between occasions within students. This variance components model is not interesting in itself but it provides a baseline with which to compare more complex models. We define the model and display it in the **Equations** window as follows:


Equations
reading$_{ij}$ ~ N(XB, Ω)
reading$_{ij}$ = β_{0ij}cons
$\beta_{0ij} = \beta_0 + u_{0j} + e_{0ij}$
$\begin{bmatrix} u_{0j} \end{bmatrix}$ ~ N(0, Ω_u) : $\Omega_u = \begin{bmatrix} \sigma^2_{u0} \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix}$ ~ N(0, Ω_e) : $\Omega_e = \begin{bmatrix} \sigma^2_{e0} \end{bmatrix}$
(1758 of 2442 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

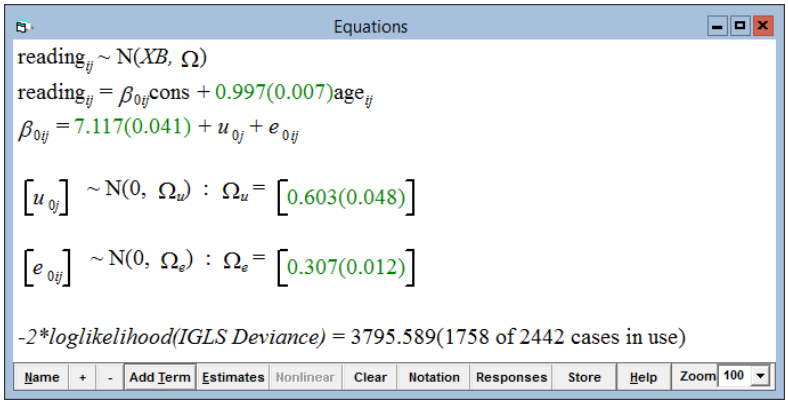
At convergence the estimates are:


Equations
reading$_{ij}$ ~ N(XB, Ω)
reading$_{ij}$ = β_{0ij}cons
β_{0ij} = 7.115(0.053) + u_{0j} + e_{0ij}
$\begin{bmatrix} u_{0j} \end{bmatrix}$ ~ N(0, Ω_u) : $\Omega_u = \begin{bmatrix} 0.078(0.083) \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix}$ ~ N(0, Ω_e) : $\Omega_e = \begin{bmatrix} 4.561(0.172) \end{bmatrix}$
-2*loglikelihood(IGLS Deviance) = 7685.737(1758 of 2442 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

As we would expect, given the way we have defined our response, the variation between occasions within students is large and overwhelms the variation between students. The likelihood statistic (-2 loglikelihood), found at the bottom of the **Equations** window, can be used as the basis for judging more elaborate models. The baseline value is 7685.7.

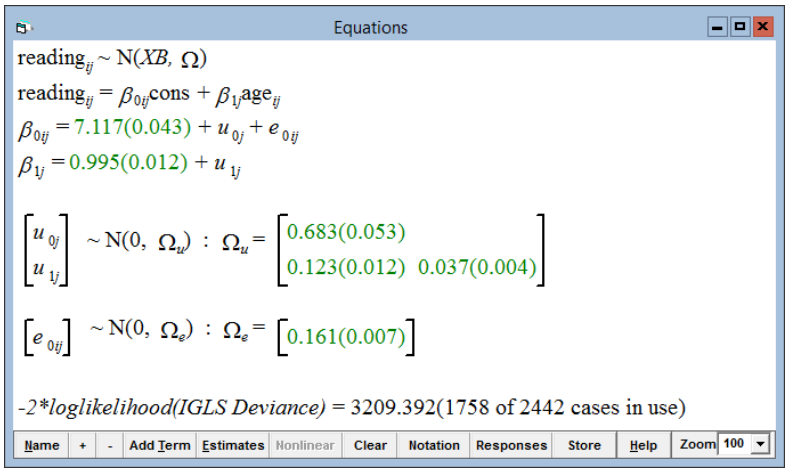
13.3 A linear growth curve model

A first step in modelling the between-occasion within-student, or level 1, variation would be to fit a fixed linear trend. We therefore add **age** to our list of fixed explanatory variables in the **Equations** window, click on **More** and at convergence obtain the following:


Equations
reading$_{ij}$ ~ N(XB, Ω)
reading$_{ij}$ = β_{0ij}cons + 0.997(0.007)age$_{ij}$
β_{0ij} = 7.117(0.041) + u_{0j} + e_{0ij}
$\begin{bmatrix} u_{0j} \end{bmatrix}$ ~ N(0, Ω$_u$) : Ω$_u$ = $\begin{bmatrix} 0.603(0.048) \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix}$ ~ N(0, Ω$_e$) : Ω$_e$ = $\begin{bmatrix} 0.307(0.012) \end{bmatrix}$
-2*loglikelihood(IGLS Deviance) = 3795.589(1758 of 2442 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

The estimate of the fixed parameter for **age** is very close to 1 because of the way the scale is defined. We see marked changes from our previous model in the estimates of the random parameters. We get an estimate of the level 2 variance that is about twice the size of the remaining level 1 variance, and a large reduction in the likelihood statistic, which is now 3795.6.

We would expect the linear growth rate to vary from student to student around its mean value of 1, rather than be fixed, and so we make the coefficient of **age** random at level 2 and continue iterations until convergence to give:


Equations
reading$_{ij}$ ~ N(XB, Ω)
reading$_{ij}$ = β_{0ij}cons + β_{1j}age$_{ij}$
β_{0ij} = 7.117(0.043) + u_{0j} + e_{0ij}
β_{1j} = 0.995(0.012) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω$_u$) : Ω$_u$ = $\begin{bmatrix} 0.683(0.053) & \\ 0.123(0.012) & 0.037(0.004) \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix}$ ~ N(0, Ω$_e$) : Ω$_e$ = $\begin{bmatrix} 0.161(0.007) \end{bmatrix}$
-2*loglikelihood(IGLS Deviance) = 3209.392(1758 of 2442 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

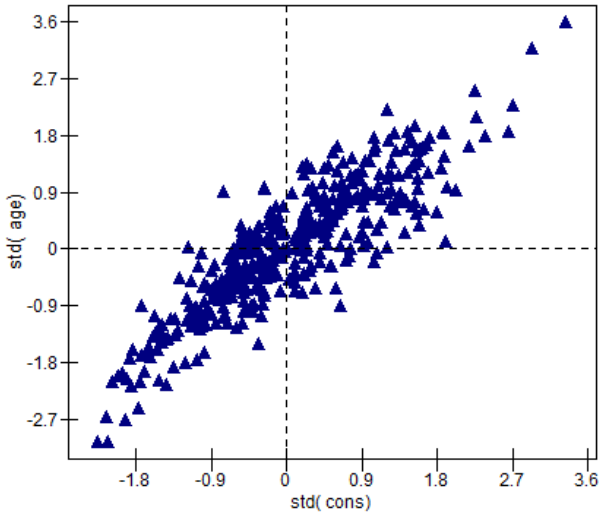
Note that the coefficient for **age** *now has a subscript j.*

The change in deviance, that is the reduction in the -2*loglikelihood statistic, is 586; this is large and is clearly statistically highly significant. Hence there is considerable variation between students in their linear growth rates. We can get some idea of the size of this variation by taking the square root of the slope variance (σ^2_{u1}) to give the estimated standard deviation (0.19). Assuming Normality, about 95% of the students will have growth rates within two standard deviations of the overall mean (= 1), giving a 95% coverage interval of 0.62 to 1.38 for the 'growth rate'.

We can also look at various plots of the level 2 residuals, using the **Residuals**

window. Below we plot the level 2 standardised residuals

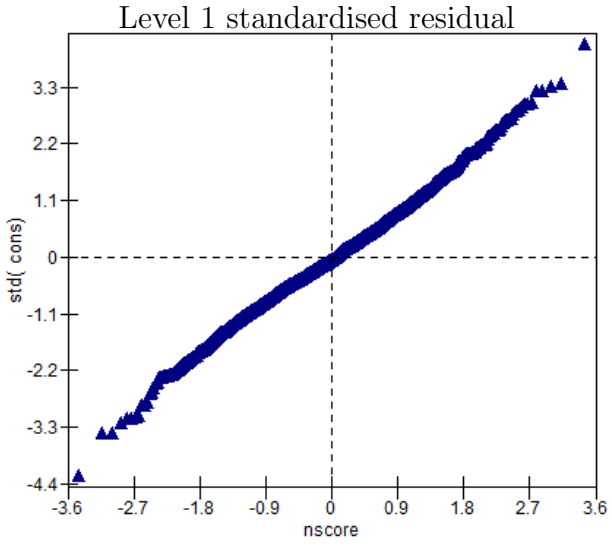
Level 2 standardised residual plot: slope vs. intercept



We see from the above plot that the two level 2 residuals are positively correlated. Using the **Estimates** window we see that the model estimate is 0.77 and shows that the greater the expected score at mean age, the faster the growth. However, this statistic needs to be interpreted with great caution: it can vary according to the scale adopted, and is relevant only for linear growth models.

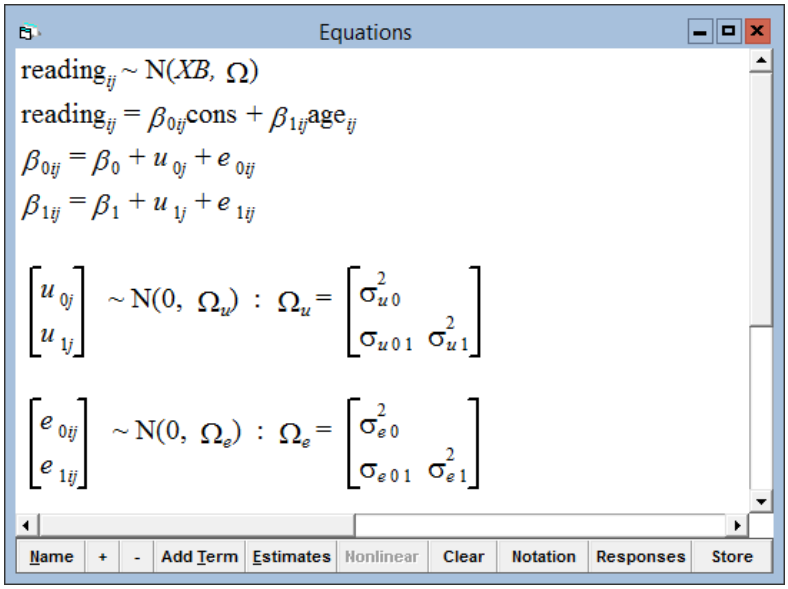
To study the distributional assumptions, we can plot the level 1 and level 2 residuals against their Normal scores: in the present case these plots conform closely to straight lines. The level 1 plot of the standardised residual against Normal score is as follows:

Level 1 standardised residual



13.4 Complex level 1 variation

Before going on to elaborate the level 2 variation, we can model complex, that is non-constant, variation at level 1 to reflect the 'constant coefficient of variation' scaling of the reading score. This requires that the total variance at each age is proportional to the square of the mean, so that we would expect both the level 2 and level 1 variances to be non-constant. To allow the level 1 variance to be a quadratic function of the predicted value, we declare the coefficient of **age** to be random at level 1 (see for example Goldstein (2003), Chapter 3). The **Equations** window is:


Equations
$\text{reading}_{ij} \sim N(XB, \Omega)$
$\text{reading}_{ij} = \beta_{0ij}\text{cons} + \beta_{1ij}\text{age}_{ij}$
$\beta_{0ij} = \beta_0 + u_{0j} + e_{0ij}$
$\beta_{1ij} = \beta_1 + u_{1j} + e_{1ij}$
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u01} & \sigma^2_{u1} \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \\ e_{1ij} \end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} \sigma^2_{e0} & \\ \sigma_{e01} & \sigma^2_{e1} \end{bmatrix}$
Name
+
-
Add Term
Estimates
Nonlinear
Clear
Notation
Responses
Store

From the **Variance function** window we see that the level 1 variance is the following function of the level 1 parameters, whose estimates are obtained by running the model to convergence:

$$(e_{0ij}\text{cons} + e_{1ij}\text{age}_{ij}) = \sigma^2_{e0}\text{cons}^2 + 2\sigma_{e01}\text{cons} \times \text{age}_i j + \sigma^2_{e1}\text{age}^2_{ij}$$

As a result of allowing the level 1 variance to have this form, there is a statistically significant decrease in the likelihood statistic of 32.4 with 2 degrees of freedom. We shall see later that some of this level 1 variation can be explained by further modelling of the level 2 variation.

13.5 Repeated measures modelling of non-linear polynomial growth

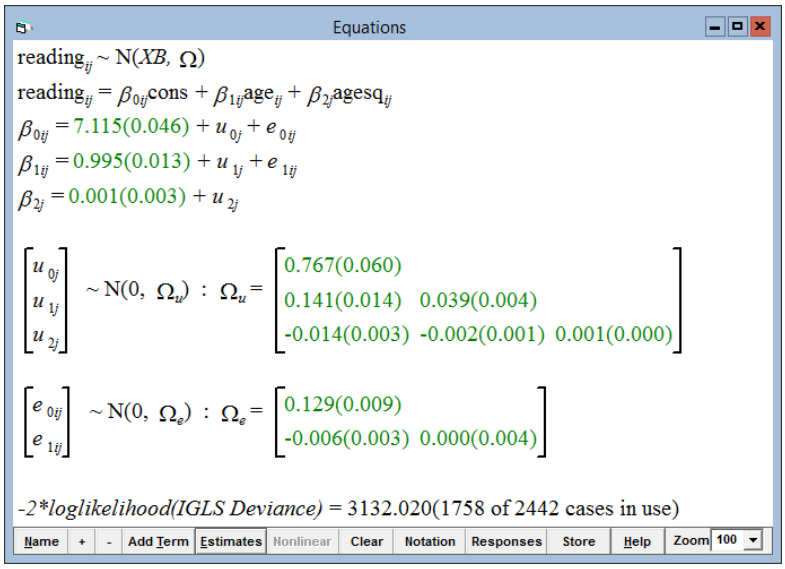
Growth in reading may not be linear for all students over this age range. One simple way of inducing nonlinearity is to define a quadratic term in age.

- Type the following in the bottom box of the **Command interface** window and press return:

```
► calc c19 = 'age'^2
```

- Use the **Names** window to assign the name **agesq** to **c19**

Add **agesq** to the model in the fixed part with a coefficient random at the student level. At convergence we have:

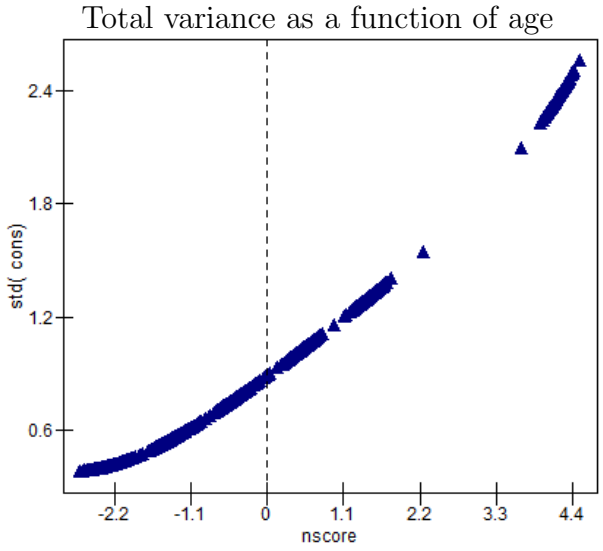

Equations
reading$_{ij}$ ~ N(XB, Ω)
reading$_{ij}$ = β_{0ij}cons + β_{1ij}age$_{ij}$ + β_{2j}agesq$_{ij}$
β_{0ij} = 7.115(0.046) + u_{0j} + e_{0ij}
β_{1ij} = 0.995(0.013) + u_{1j} + e_{1ij}
β_{2j} = 0.001(0.003) + u_{2j}
~ N(0, Ω_u) : Ω_u =
0.767(0.060)
0.141(0.014) 0.039(0.004)
-0.014(0.003) -0.002(0.001) 0.001(0.000)
~ N(0, Ω_e) : Ω_e =
0.129(0.009)
-0.006(0.003) 0.000(0.004)
-2*loglikelihood(IGLS Deviance) = 3132.020(1758 of 2442 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

The likelihood statistic shows a further drop, this time by 45 with 4 degrees of freedom (one fixed parameter and three random parameters), so there is strong evidence that a quadratic term, which varies from student to student, improves the model.

Note that the fixed parameter for agesq is very small because of the way the scale was defined over age. The level 1 random parameter estimates are different from those of the previous model. The σ^2_{e1} parameter is extremely small and nonsignificant, and we estimate a linear effect with age for the between-occasion variance. The display precision in the above window is 3 significant digits after the decimal point. If this is increased to 4 (use the Display precision item on the Options menu) we see that the estimate is actually 0.0004

What has happened is that the more complex level 2 variation which we have introduced in order to model nonlinear growth in individuals has absorbed much of the residual level 1 variation in the earlier model. We can view this final model for the random variation as a convenient and reasonably parsimonious description of how the overall variance produced by the assumption

of a constant coefficient of variation is partitioned between the levels. We can use the **Variance function** window to calculate the variance at both level 1 and level 2 for each record in the data set. If we place these into separate columns (say, **c28** and **c29**) and then add the two columns together into, say, **c30** using the **Calculate** window, we obtain the total predicted variance. The plot below shows this as a function of age, confirming the original definition of the variance as a quadratic function of the mean, which itself is defined to be a linear function of age.

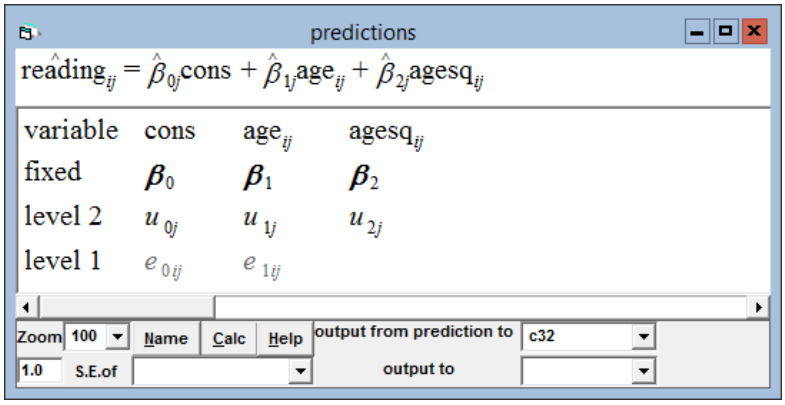


Since the overall relationship of the mean and variance with age is to some extent arbitrary and our choice, our principal interest lies in how the growth of individuals varies. The model parameters and derived variance functions describe these growth patterns, and we can also display the estimated growth lines for selected individuals or groups. For example, to plot the lines for the first four individuals, let us set up a *filter* column, say **c31**, which is 1 if the record belongs to one of these individuals and zero otherwise. This is achieved by typing in the **Calculate** window:

```
► c31 = 'student' < 5
```

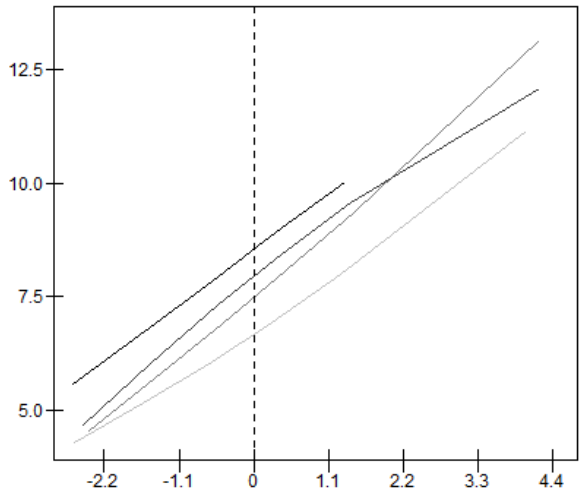
See the **Help** system for a detailed description of how to use this window.

Now open the **Predictions** window and compute predicted values using the fixed part coefficients plus the level 2 random coefficients, placing the result into column 32, as follows:


predictions
$\hat{reading}_{ij} = \hat{\beta}_{0j}cons + \hat{\beta}_{1j}age_{ij} + \hat{\beta}_{2j}agesq_{ij}$
variable cons age_{ij} $agesq_{ij}$
fixed β_0 β_1 β_2
level 2 u_{0j} u_{1j} u_{2j}
level 1 e_{0ij} e_{1ij}
Zoom 100 Name Calc Help output from prediction to c32
1.0 S.E.of output to

- Open the **Customised graph** window and make the following selections on the **plot what?** tab: **c32** for **y**, **age** for **x**, **c31** as the **filter**, **student** as the **group** variable and **line** as the **plot type**
- In the **colour** selector on the **plot style** tab, choose **16 rotate**
- Click **Apply**

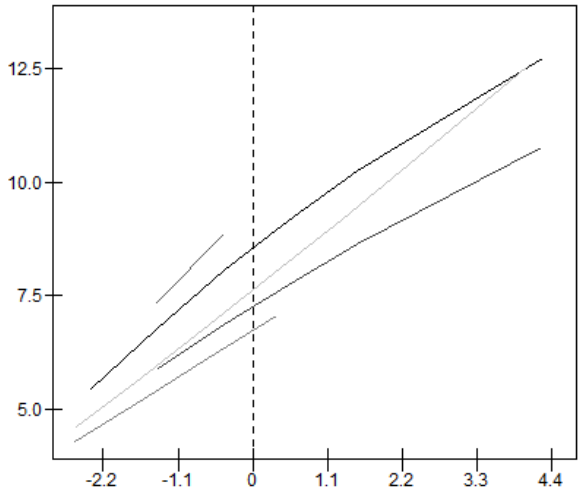
The following plot will appear, with each student represented by a line.



We can easily display other student lines by redefining **c31** with the following calculation:

```
► c31 = 'student' >= 10 & 'student' < 15
```

This immediately updates the graph to display the predicted lines for students 11 through 15 as follows, where one student only has two measurements:



We can set up quite general filter functions using the **Calculate** window, allowing us to explore the data extensively.

Various extensions are available. We can fit multivariate repeated measures models using the **Multivariate** model definition window as described in a later chapter, or extend the level 1 component to have a serial correlation structure (see Goldstein et al. (1994); for details of how to do this in MLwiN see Section 5 of the MLwiN Version 2.10 Manual Supplement).

Chapter learning outcomes

- ⋆ How to formulate repeated measures models
- ⋆ How to fit growth curve models of increasing complexity

Chapter 14

Multivariate Response Models

14.1 Introduction

Multivariate response data are conveniently incorporated into a multilevel model by creating an extra level "below" the original level 1 units to define the multivariate structure. We thus have responses within individuals that are in turn nested within higher-level units.

This chapter will show how to specify and fit a relatively straightforward multivariate model with Normal responses. We shall briefly deal with the case of multivariate response models for categorical response variables.

The example data set

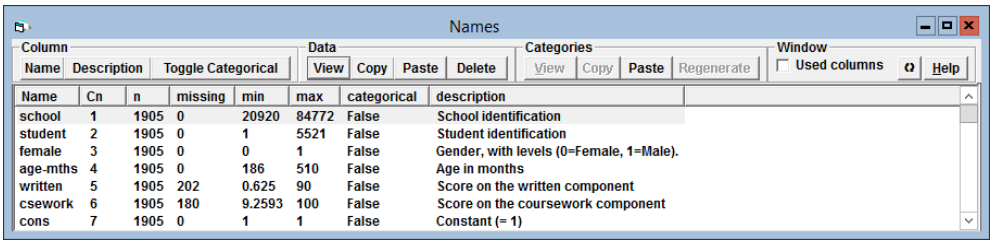
We shall be using data consisting of scores on two components of a science examination taken in 1989 by 1905 students in 73 schools in England. The examination is the General Certificate of Secondary Education (GCSE) taken at the end of compulsory schooling, normally when students are 16 years of age. The first component is a traditional written question paper (marked out of a total score of 160), and the second consists of coursework (marked out of a total score of 108), including projects undertaken during the course and marked by each student's own teacher (but 'moderated', i.e., a sample is checked by external examiners). Both components' scores have been rescaled so that their maximum is 100.

Interest in these data centres on the relationship between the component marks at both the school and student level, whether there are gender differences in this relationship and whether the variability differs for the two components.

Open the worksheet **gcsemv1.ws** supplied with the MLwiN software. The variables in the worksheet and shown in the **Names** window are defined as follows:

Variable	*Description*
school	School identification
student	Student identification
female	1 if female, 0 if male
age-mths	Age in months
written	Score on the written component
csework	Score on the coursework component
cons	Constant (= 1)

The Names window also shows that the two response variables each have approximately 10% missing, so that about 20% of students have a single response. For present purposes we assume that missing is completely at random.



Name	Cn	n	missing	min	max	categorical	description
school	1	1905	0	20920	84772	False	School identification
student	2	1905	0	1	5521	False	Student identification
female	3	1905	0	0	1	False	Gender, with levels (0=Female, 1=Male).
age-mths	4	1905	0	186	510	False	Age in months
written	5	1905	202	0.625	90	False	Score on the written component
csework	6	1905	180	9.2593	100	False	Score on the coursework component
cons	7	1905	0	1	1	False	Constant (= 1)

14.2 Specifying a multivariate model

To define a multivariate model—in our case, a bivariate model—we treat the individual student as a level 2 unit and the 'within-student' measurements as level 1 units. Each level 1 measurement 'record' has a response, which is either the written paper score or the coursework score. The basic explanatory variables are a set of dummy variables that indicate which response variable is present. Further explanatory variables are defined by multiplying these dummy variables by individual-level explanatory variables, for example gender.

Omitting school identification, the form of the data matrix is displayed in Table 14.1 for three students. Two of them have both measurements, and the third has only the coursework paper score. The first and second students are female (1), and the third is male (0).

The model for the two-level case (i.e., ignoring school) can be written as

Table 14.1: Data matrix for examination data

		Intercepts		Gender	
Student	Response	Written	Coursework	Written. gender	Coursework. gender
1 (female)	y_{11}	1	0	1	0
1	y_{21}	0	1	0	1
2 (female)	y_{12}	1	0	1	0
2	y_{22}	0	1	0	1
3 (male)	y_{13}	0	1	0	0

follows:

$$y_{ij} = \beta_0 z_{1ij} + \beta_1 z_{2ij} + \beta_2 z_{1ij} x_j + \beta_3 z_{2ij} x_j + u_{1j} z_{1ij} + u_{2j} z_{2ij}$$
$$z_{1ij} = \left\{ \begin{array}{c} 1 \text{ if written} \\ 0 \text{ if coursework} \end{array} \right\}, \quad z_{2ij} = 1 - z_{1ij}, \quad x_j = \left\{ \begin{array}{c} 1 \text{ if female} \\ 0 \text{ if male} \end{array} \right\}$$
$$\text{var}(u_{1j}) = \sigma^2_{u1}, \quad \text{var}(u_{2j}) = \sigma^2_{u2}, \quad \text{cov}(u_{1j} u_{2j}) = \sigma_{u12}$$

Alternatively we can write:

$$\begin{aligned} \mathbf{response}_{1j} &= b_{0j}\mathbf{written}_{ij} + b_2\mathbf{written.gender}_{ij} \\ b_{0j} &= b_0 + u_{0j} \\ \mathbf{response}_{2j}j &= b_{1j}\mathbf{coursework}_{ij} + b_3\mathbf{coursework.gender}_{ij} \\ b_{1j} &= b_1 + u_{1j} \\ \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} &\sim (\mathrm{N}, \Omega_u), \quad \Omega_u = \begin{bmatrix} \sigma^2_{u0} & \\ \sigma^2_{u01} & \sigma^2_{u1} \end{bmatrix} \end{aligned} \tag{14.1}$$

where $\mathbf{response}_{1j}$ is the written score for student j and $\mathbf{response}_{2j}$ is the coursework score for student j.

There are several interesting features of this model. There is no level 1 variation specified because level 1 exists solely to define the multivariate structure. The level 2 variances and covariance are the (residual) between-student variances. In the case where only the intercept dummy variables are fitted, and in the case where every student has both scores, the model estimates of these parameters become the usual between-student estimates of the variances and covariance. The multilevel estimates are statistically efficient even where some responses are missing, and in the case where the measurements have a multivariate Normal distribution IGLS provides maximum likelihood estimates.

Thus, the formulation as a 2-level model allows for the efficient estimation of a covariance matrix with missing responses, where the missingness is at random. This means, in particular, that studies can be designed in such a way that not every individual has every measurement, with measurements randomly allocated to individuals. Such 'rotation' or 'matrix' designs are

common in many areas and may be efficiently modelled in this way. A more detailed discussion is given by Goldstein (2003) (Chapter 4). Furthermore, the ability to provide estimates of covariance matrices at each higher level of a data hierarchy enables us to conduct additional forms of modelling such as multilevel factor analysis (see Rowe & Hill (1998).

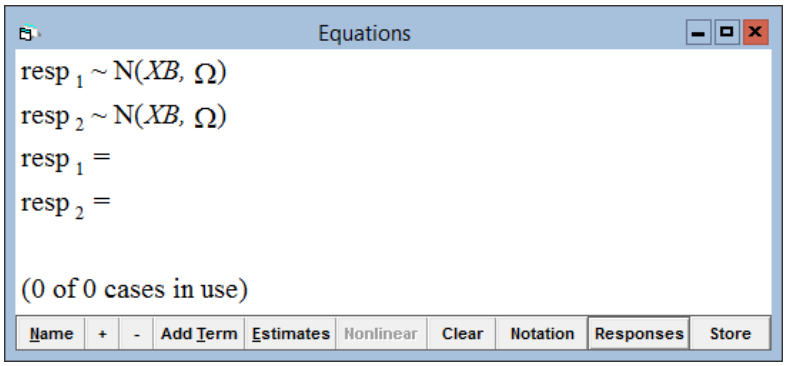
A third level (school) can be incorporated, and this is specified by inserting a third subscript, k, and two associated random intercept terms.

14.3 Setting up the basic model

Now let's set up our first model. Begin by opening the **Equations** window. We need to tell the software we have more than one response variable.

- Click on the **Responses** button on the **Equations** window's toolbar
- In the **Specify responses** window that appears, highlight **written** and **csework** and click **Done**

Two things happen. Firstly, the **Equations** window is updated



We can see the **Equations** window starting to take the form of equation (14.1). Secondly, if we look at the **Names** window we see two new variables have been created, **resp** and **resp_indicator**. The former contains the stacked responses and the latter is a categorical variable indicating which response the current data row applies to.

We now need to specify that responses 1 and 2 are nested within students

- Click on either **resp$_1$ or resp$_2$**

Notice that so far we have a one-level model set up with level 1 defined by the **response_indicator** column.

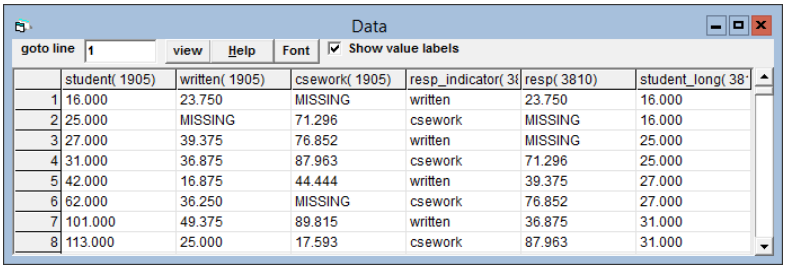
- In the **Y variable** window, set **N levels** as **2-ij**
- Set **level 2(j)** as **student**
- Click **Done**

If you click on $\mathbf{resp}_1$ again you will see that **student** has been replaced by a newly created column called **student_long**. Let's look at the original response columns and the created data:

- From the **Data Manipulation** menu, select **View or edit data**
- Click on the **view** button in the **Data** window
- Select **student, written, csework, resp_indicator, resp** and **student_long**
- Click **OK**

We see the original student identifier and response columns have a length of 1905 with one row per child. The newly created columns are exactly twice as long, 3810, with one row per response.

Note the way that MLwiN treats the missing responses for students 16 and 25.

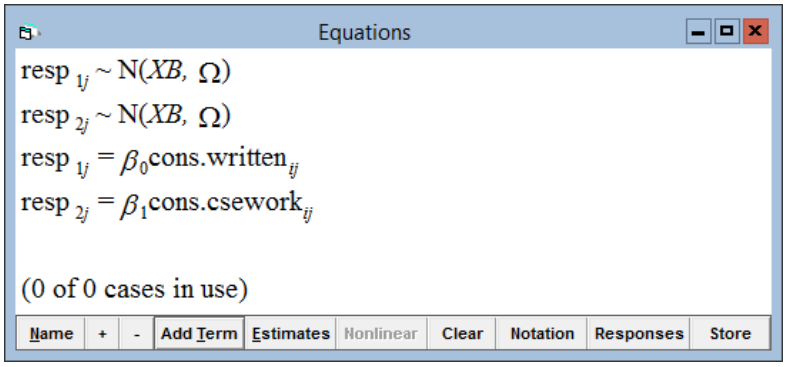


	student(1905)	written(1905)	csework(1905)	resp_indicator(3	resp(3810)	student_long(38
1	16.000	23.750	MISSING	written	23.750	16.000
2	25.000	MISSING	71.296	csework	MISSING	16.000
3	27.000	39.375	76.852	written	MISSING	25.000
4	31.000	36.875	87.963	csework	71.296	25.000
5	42.000	16.875	44.444	written	39.375	27.000
6	62.000	36.250	MISSING	csework	76.852	27.000
7	101.000	49.375	89.815	written	36.875	31.000
8	113.000	25.000	17.593	csework	87.963	31.000

Now let's add explanatory variables into the model. We begin by adding two intercepts, one for the written response and one for the coursework response:

- Click on the **Add Term** button in the **Equations** window
- In the **Specify Term** window that appears, select **cons** from the **variable** drop-down list
- Click the **add Separate coefficients** button

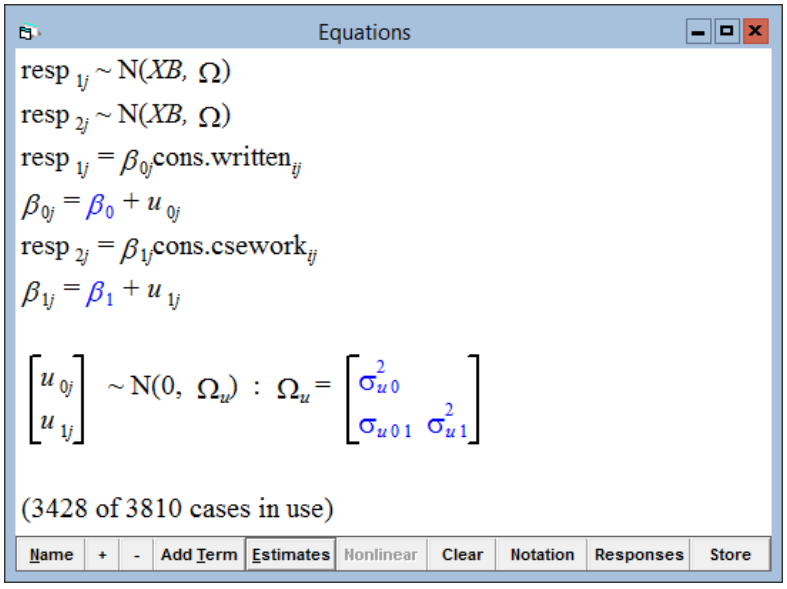
The **Equations** window becomes:


Equations
resp 1j ~ N(XB, Ω)
resp 2j ~ N(XB, Ω)
resp 1j = β0cons.written ij
resp 2j = β1cons.csework ij
(0 of 0 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

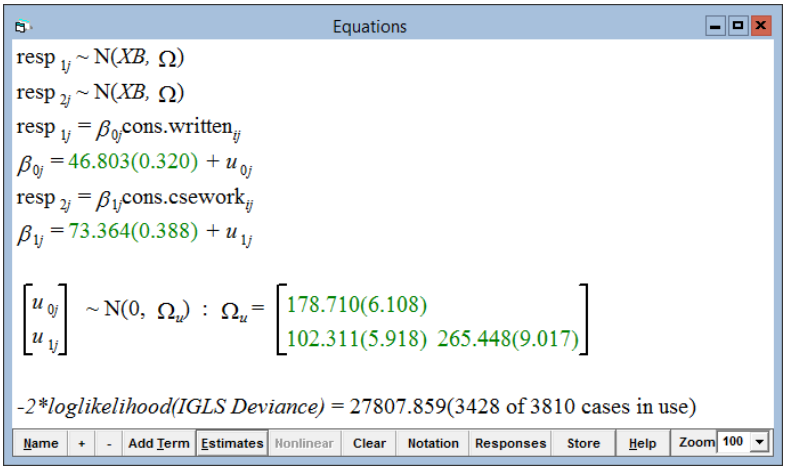
To complete the model specification we need to specify a 2×2 covariance matrix of the responses at the student level. To do this

- Click on β_0
- In the **X variable** window, check the **j(student_long)** check box and click **Done**
- Repeat the procedure for β_1
- Click on the **Estimates** button

The **Equations** window now looks like this:


Equations
resp 1j ~ N(XB, Ω)
resp 2j ~ N(XB, Ω)
resp 1j = β0jcons.written ij
β0j = β0 + u 0j
resp 2j = β1jcons.csework ij
β1j = β1 + u 1j
[u 0j; u 1j] ~ N(0, Ωu) : Ωu = [σ²u0 ; σu01 σ²u1]
(3428 of 3810 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The window shows the same model structure as equation (14.1). If we run the model by clicking **Start** and then **Estimates** we see:

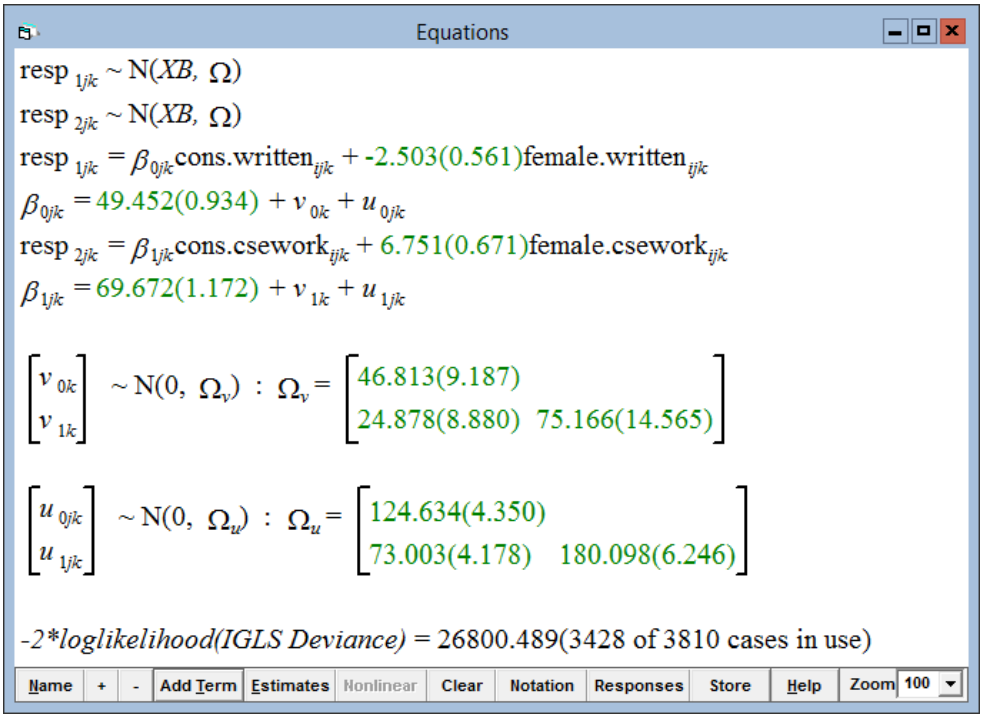

Equations
resp 1j ~ N(XB, Ω)
resp 2j ~ N(XB, Ω)
resp 1j = β0j cons.written ij
β0j = 46.803(0.320) + u 0j
resp 2j = β1j cons.csework ij
β1j = 73.364(0.388) + u 1j
[u 0j; u 1j] ~ N(0, Ωu) : Ωu = [178.710(6.108); 102.311(5.918) 265.448(9.017)]
-2*loglikelihood(IGLS Deviance) = 27807.859(3428 of 3810 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

Here we estimate the two means and the covariance matrix for the two responses. The advantage of fitting this model in a multilevel framework is that we do not have to delete cases where one of the responses is missing.

Let's now elaborate the model by partitioning the covariance matrix into between-student and between-school components. We also include gender effects in the fixed part of the model.

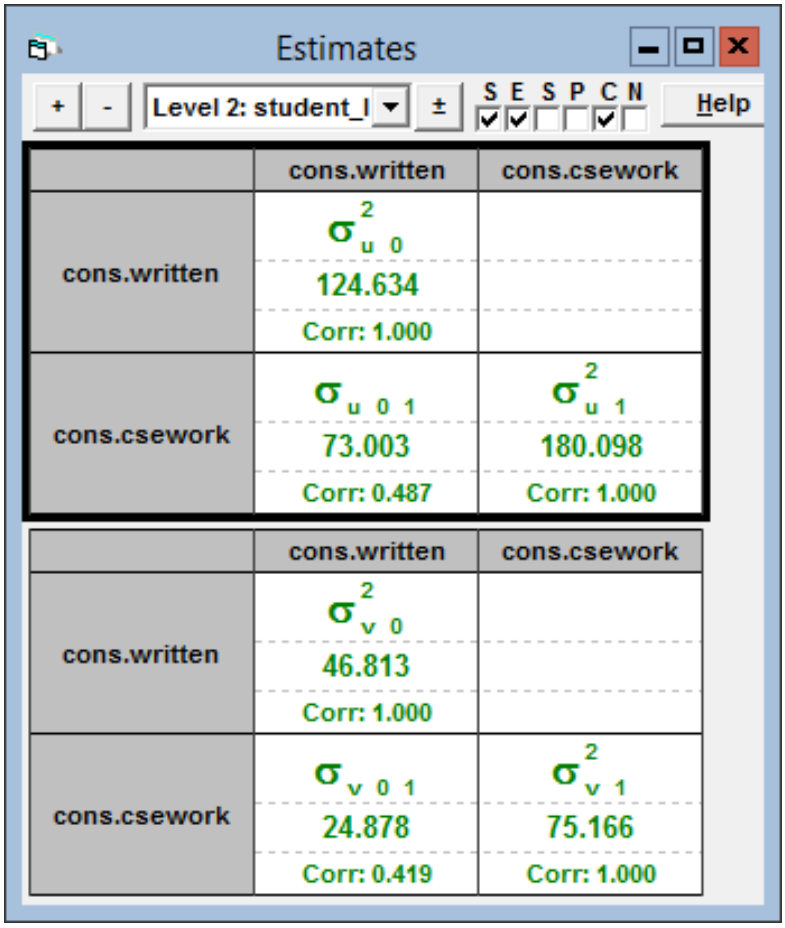
- Click on **resp** (1 or 2)
- In the **Y variable** window, set **N levels:** as **3-ijk**
- Set **level 3(k):** as **school** and click **done**
- Click on β_0 and β_1 in turn, and in the **X variable** window, check the **k(school_long)** check box (clicking **Done** each time)
- Click on the **Add Term** button
- Select **female** from the **variable** drop-down list
- Click the **add Separate coefficients** button
- Run the model by clicking **Start**

The **Equations** window should now look like this:


Equations
resp 1jk ~ N(XB, Ω)
resp 2jk ~ N(XB, Ω)
resp 1jk = β0jk cons.written ijk + -2.503(0.561) female.written ijk
β0jk = 49.452(0.934) + v 0k + u 0jk
resp 2jk = β1jk cons.csework ijk + 6.751(0.671) female.csework ijk
β1jk = 69.672(1.172) + v 1k + u 1jk
[v 0k, v 1k] ~ N(0, Ωv) : Ωv = [46.813(9.187); 24.878(8.880) 75.166(14.565)]
[u 0jk, u 1jk] ~ N(0, Ωu) : Ωu = [124.634(4.350); 73.003(4.178) 180.098(6.246)]
-2*loglikelihood(IGLS Deviance) = 26800.489(3428 of 3810 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

The coefficients for **female.written** (-2.5) and **female.csework** (6.8) tell us the gender difference for the written and coursework components, respectively. Both coefficients are statistically significant. The girls do somewhat worse than the boys on the written paper but considerably better on the coursework component. The coursework component also has a larger variance at both the student and school levels. The correlations between the coursework and written scores are 0.42 and 0.49 at school and student level respectively. The intra-school correlation is 0.27 for the written paper and 0.29 for the coursework.

We can often view the results more conveniently using the **Estimates** window, which is opened by selecting **Estimate tables** from the **Model** menu. Use the **Help** button on this window to obtain details on how to manipulate the display. Below is an example of a display of the level 2 and level 3 random parameters matrices, showing a symbol, an estimate and a correlation for each element.



14.4 A more elaborate model

We now let the coefficient of gender (**female**) be random at the school level.

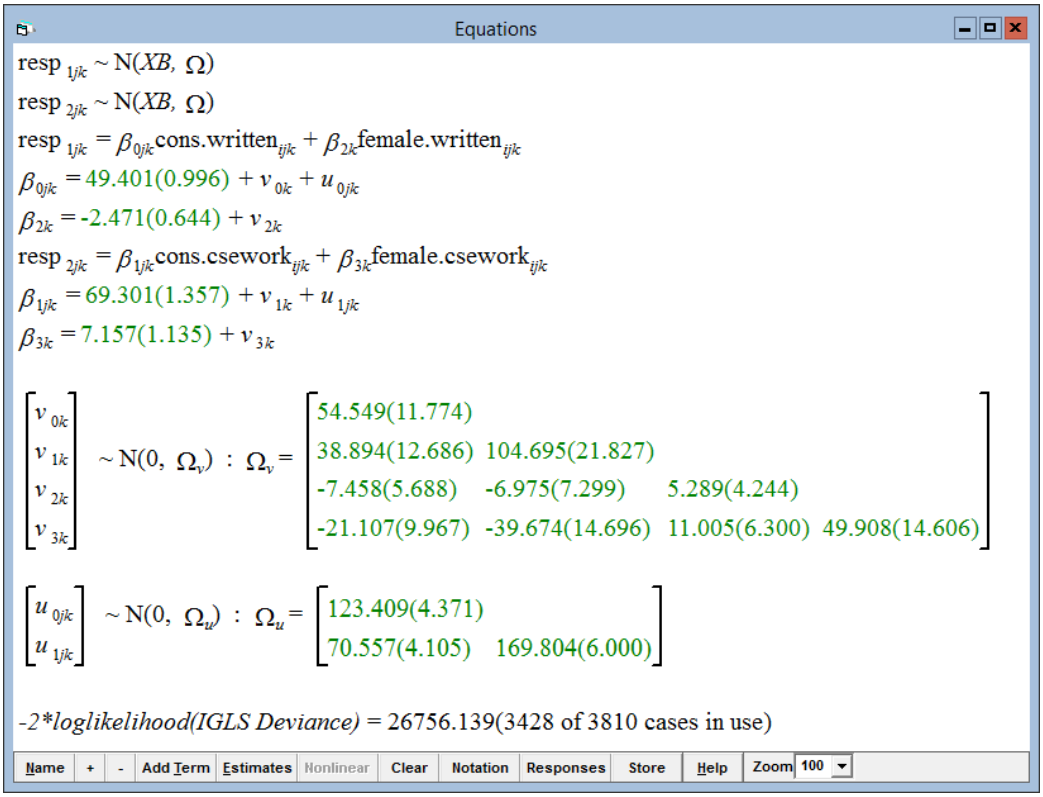
- Click on **female.written** and **female.csework** in turn, and in the **X variable** window, check the **k(school_long)** check box (clicking **Done** each time)

We get the estimates shown below and focus on the school-level covariance matrix.

Note the numbering of the random error terms to correctly identify the parameters in this matrix. Here, the first and second terms are for the written and coursework intercepts, respectively, and the third and fourth are for the written and coursework gender differences.

The value of the likelihood statistic is 26756.1, so that the deviance statistic (relative to the previous model) is 44.4 with 7 degrees of freedom; this is highly significant. The variance of the coursework gender difference is rather

large. A variance of 49.9 implies a 95% coverage range of plus or minus 15 points around the average difference of 7.2. We notice, however, that the estimate for the variance of the gender difference for the written paper is close to its standard error, as are the covariances with the gender difference.

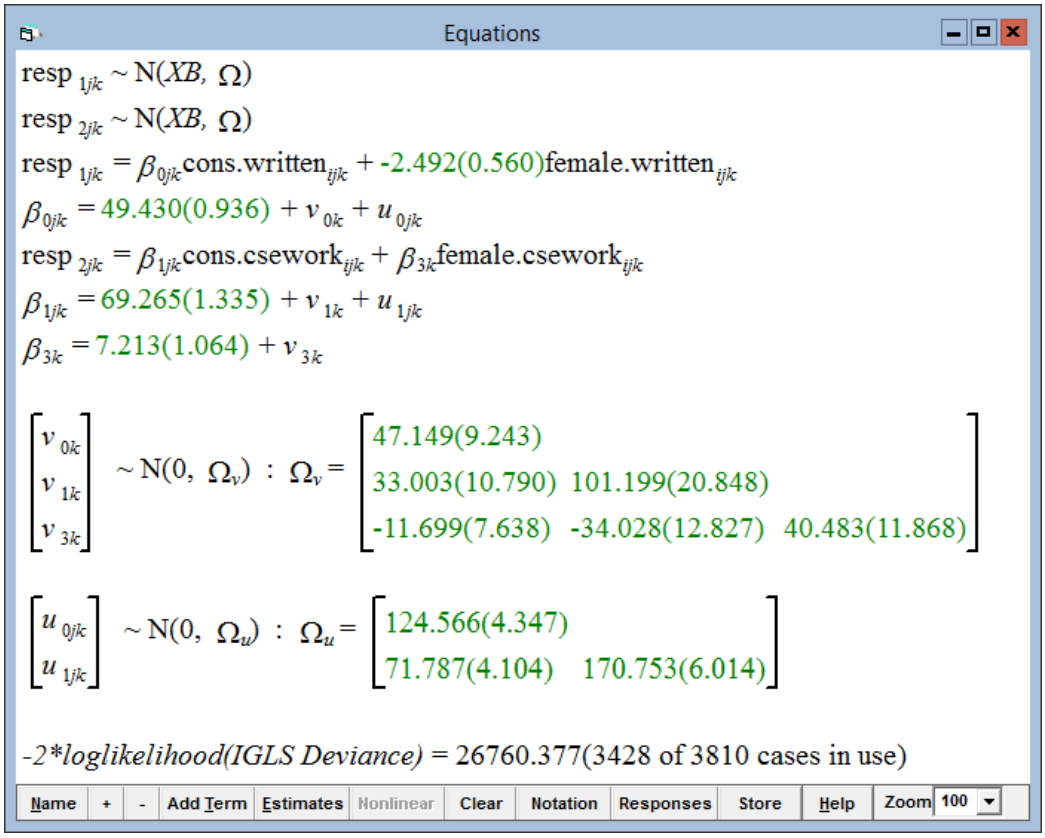

Equations
resp 1jk ~ N(XB, Ω)
resp 2jk ~ N(XB, Ω)
resp 1jk = β0jk cons.written ijk + β2k female.written ijk
β0jk = 49.401(0.996) + v 0k + u 0jk
β2k = -2.471(0.644) + v 2k
resp 2jk = β1jk cons.csework ijk + β3k female.csework ijk
β1jk = 69.301(1.357) + v 1k + u 1jk
β3k = 7.157(1.135) + v 3k
v 0k v 1k v 2k v 3k ~ N(0, Ωv) : Ωv = 54.549(11.774) 38.894(12.686) 104.695(21.827) -7.458(5.688) -6.975(7.299) 5.289(4.244) -21.107(9.967) -39.674(14.696) 11.005(6.300) 49.908(14.606)
u 0jk u 1jk ~ N(0, Ωu) : Ωu = 123.409(4.371) 70.557(4.105) 169.804(6.000)
-2*loglikelihood(IGLS Deviance) = 26756.139(3428 of 3810 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

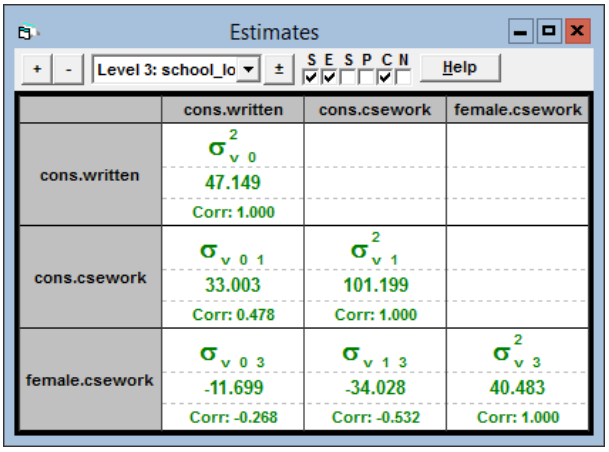
Now fit the model with the random term for **female.written** omitted.

- Click on **female.written**
- In the **X variable** window, uncheck the **k(school_long)** check box and click **Done**

The results are shown in the upper figure on the following page.

The likelihood statistic is now 26760.4 so that, compared with the preceding model, the deviance is only 4.3 with four degrees of freedom. Thus, while there is variation among schools in the gender difference for the coursework component, there is no evidence that there is any such gender-related variation for the written paper. The correlations are shown in the matrix in the lower figure on the next page.


Equations
resp 1jk ~ N(XB, Ω)
resp 2jk ~ N(XB, Ω)
resp 1jk = β0jk cons.written ijk + -2.492(0.560) female.written ijk
β0jk = 49.430(0.936) + v 0k + u 0jk
resp 2jk = β1jk cons.csework ijk + β3k female.csework ijk
β1jk = 69.265(1.335) + v 1k + u 1jk
β3k = 7.213(1.064) + v 3k
v 0k v 1k v 3k ~ N(0, Ωv) : Ωv = 47.149(9.243) 33.003(10.790) 101.199(20.848) -11.699(7.638) -34.028(12.827) 40.483(11.868)
u 0jk u 1jk ~ N(0, Ωu) : Ωu = 124.566(4.347) 71.787(4.104) 170.753(6.014)
-2*loglikelihood(IGLS Deviance) = 26760.377(3428 of 3810 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100



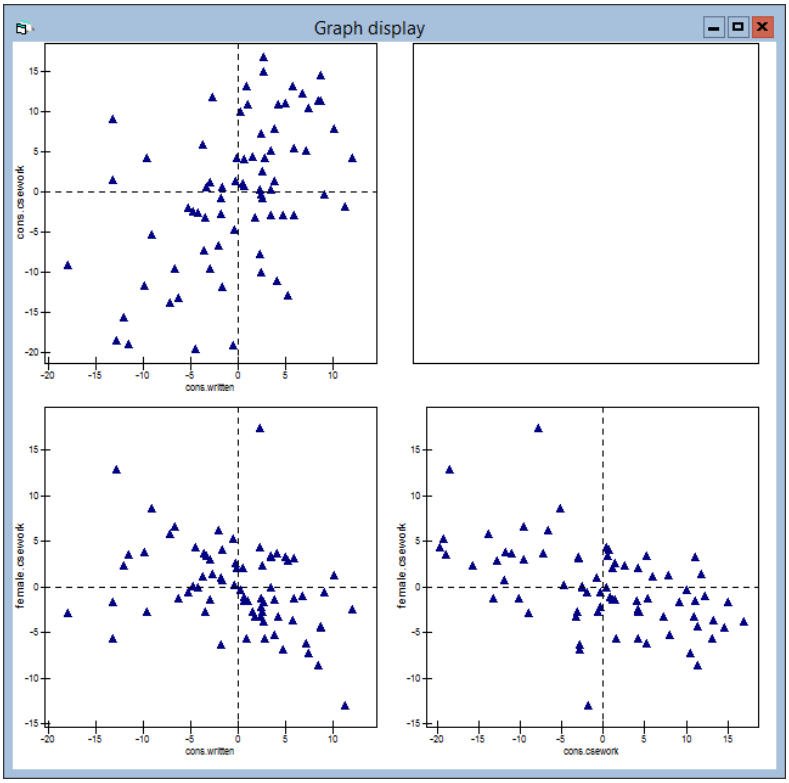
The correlation between the school-level, gender difference coursework residual and the school-level coursework residual for the intercept (i.e., the mean for boys) is $\rho(v_3, v_1) = -0.531$. This indicates that in schools which have high means for boys on coursework (a positive v_1 residual) the gender difference will tend to be negative (a negative v_3 residual). In other words, in such schools, girls will tend to score worse than boys. Conversely, schools that have a low mean for boys will tend to have a positive gender difference. If we look at the pairwise residual plots, we can observe this pattern.

- Select **Residuals** from the **Model** menu to open the **Residuals** window
- From the **level:** drop-down list on the **Settings** tab, select

3:school_long

- Click the **Calc** button
- Select the **Plots** tab of the **Residuals** window
- Select the **residuals** option in the **pairwise** frame
- Click the **Apply** button

The following trellis graph of pairwise plots of the school level residuals appears.



In the bottom right graph we see the plot showing the negative correlation between male coursework residuals and gender difference residuals.

14.5 Multivariate models for discrete responses

We will quickly illustrate the use of multivariate multilevel models for discrete responses and the interpretation of between-response covariances at the lowest level. To do this let's work with the data set we used in Chapter 2.

- Use **Open worksheet** on the **File** menu to open the tutorial worksheet file **tutorial.ws**

Let's make binary variables from **normexam** and **standlrt**.

- On the **File** menu select **New Macro**
- Type the following commands into the macro editor window that appears (called **Untitled:1**):

```
► chan 0 4 c3 1 c11
► chan -4 0 c11 0 c11
► chan 0 4 c5 1 c12
► chan -4 0 c12 0 c12
► name c11 'binexam' c12 'binlrt'
```

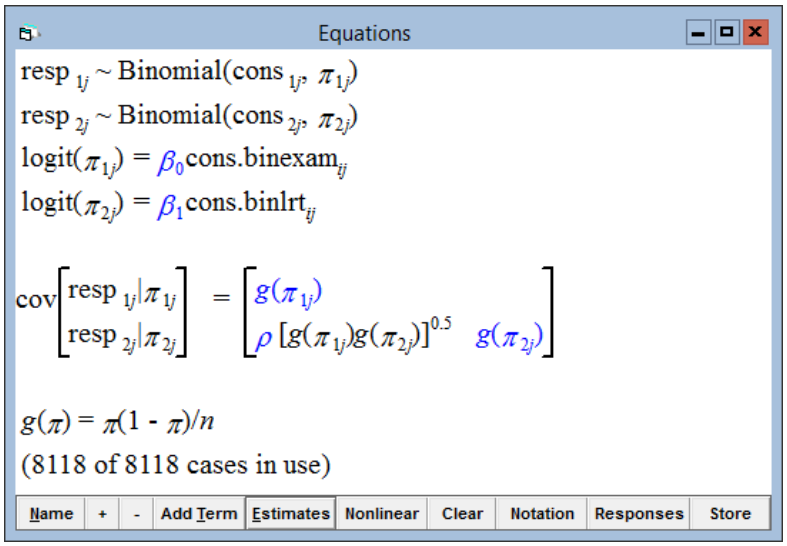
- Click the **Execute** button on the macro editor window

*Note that you could have instead used the **Recode Variables** window (accessed from the **Data Manipulation** menu) to dichotomise the two variables.*

Now let's set up a model where we estimate the covariance between these two binary responses.

- Click the **Responses** button in the **Equations** window toolbar
- In the **Specify responses** window, select **binexam** and **binlrt** and click **Done**
- Click on the distribution specifier **N** in the first line of the **Equations** window.
- In the **Response Type** window, select **Binomial** (and **logit**) and click **Done**
- In the same way, set the second response to be **Binomial**
- Click the **Add Term** button
- In the **Specify term** window, select **cons** and click on **add Separate coefficients**
- Click on n_1 in the **Equations** window
- In the **Specify denominator** window, select **cons** and click **Done**
- In the same way, set the denominator for the second response to be **cons**
- Click on **resp**$_1$ (or **resp**$_2$) and in the **Y variable** window set **N levels**: to be **2-ij**
- Set **level 2(j)** to be **student**, and click **done**
- Click **Estimates**

The **Equations** window should look like this:


Equations
resp 1j ~ Binomial(cons 1j, π1j)
resp 2j ~ Binomial(cons 2j, π2j)
logit(π1j) = β0cons.binexamij
logit(π2j) = β1cons.binlrtij
cov[resp 1j|π1j, resp 2j|π2j] = [g(π1j), ρ [g(π1j)g(π2j)]^0.5, g(π2j)]
g(π) = π(1 - π)/n
(8118 of 8118 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

The variance of each response is given the appropriate binomial form. We also estimate the correlation, at the student level, between these two binomial responses. After running the model, you will find this correlation to be 0.419. Remember that level 1 was set up only to accommodate the multivariate structure, so this is a single-level, unconditional model (with no missing data). We therefore get the same answer if we simply correlate the two binary responses using the traditional formula for a correlation coefficient. You can verify this using the **Averages and Correlation** window, which can be accessed from the **Basic Statistics** menu.

The advantage of fitting a multilevel model in this situation is that we can extend the model to handle missing data, extra covariates and higher-level random effects.

MLwiN has some ability to handle a mixture of response types. It can handle a mixture of any number of Normally distributed response variables with any number of Binomially distributed variables. The software can also handle a mixture of any number of Poisson variables with any number of Normal ones. Negative binomial or multinomial response variables cannot be included in multivariate response models, but can be used in univariate response models.

Chapter learning outcomes

- ⋆ An understanding of how multivariate models can be accommodated into a multilevel structure by specifying response measurements at level 1
- ⋆ How to use MLwiN to specify multilevel multivariate response models

⋆ How to interpret multilevel multivariate response models

Chapter 15

Diagnostics for Multilevel Models

15.1 Introduction

Diagnostic procedures, such as the detection of outliers and data points with a large influence on the fit of a model, are an important part of ordinary least squares regression analysis. The aim of this chapter is to demonstrate, via an example analysis, how some of the concepts and diagnostic tools used in regression modelling can be translated into the multilevel modelling situation. The statistical techniques used and the example explored in this chapter are dealt with in detail in Langford & Lewis (1998).

Data exploration techniques, including the detection of outlying observations, are a little-explored area of multilevel modelling. For ordinary regression, there is an extensive literature on the detection and treatment of single outliers, and an increasing literature on multiple outliers (Barnett & Lewis, 1994). However, in data structures of increasing complexity, the concept of an outlier becomes less clear-cut. For example, in a multilevel model structure, we may wish to know at what level(s) a particular response is outlying, and in respect to which explanatory variable(s). We use the term *level* to describe the unit of analysis in our model. In a multilevel model, more than one unit of analysis is appropriate for the data (Goldstein, 2003).

In a simple educational example, we may have data on examination results in a 2-level structure with students nested within schools, and either students or schools may be considered as being outliers at their respective levels in the model. Suppose, for example, that at the school level a particular school is found to be a discordant outlier; we will need to ascertain whether it is discordant due to a systematic difference affecting all the students measured within that school, or because one or two students are responsible for the discrepancy. At the student level, an individual may be outlying with respect

to the overall relationships found across all schools, or be unusual only in the context of his / her particular school. Indeed, these concepts become more complex when there are more than two levels.

Other concepts become similarly more complex. For example, masking of outlying observations, where the presence of one observation may conceal the importance of another (Atkinson, 1986), can apply across the levels of a model. The outlying nature of a school may be masked because of the presence of another similarly outlying school, or by the presence of a small number of students within the school whose influence brings the overall relationships within that school closer to the average for all schools. Other effects such as swamping (Barnett & Lewis, 1994) and other measures of joint or conditional influence (Lawrence, 1995) may also occur within as well as between units at the higher levels of a multilevel model.

An educational example

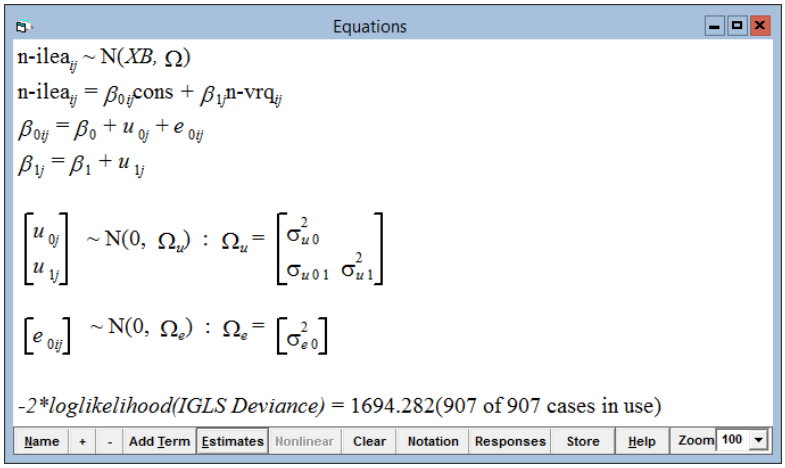
In the classic paper Aitkin & Longford (1986), the authors report an analysis of 907 students in 18 schools in a Local Education Authority in the United Kingdom. They discuss the implications of fitting different models to the data on parameter estimates and their interpretation. Of particular interest here is the presence of two single-sex grammar schools in the data, which are otherwise made up of comprehensive schools. Our analysis focuses on whether these two schools are discordant outliers in the data set, and thus of a genuinely different character. More generally, it uses the data as an example of how an examination of outliers in a two-level model may be pursued, issues not investigated by Aitkin and Longford. The numbers of students per school are given in the tables below. Schools 17 and 18 are the grammar schools.

School	1	2	3	4	5	6	7	8	9
# Pupils	65	79	48	47	66	41	52	67	49

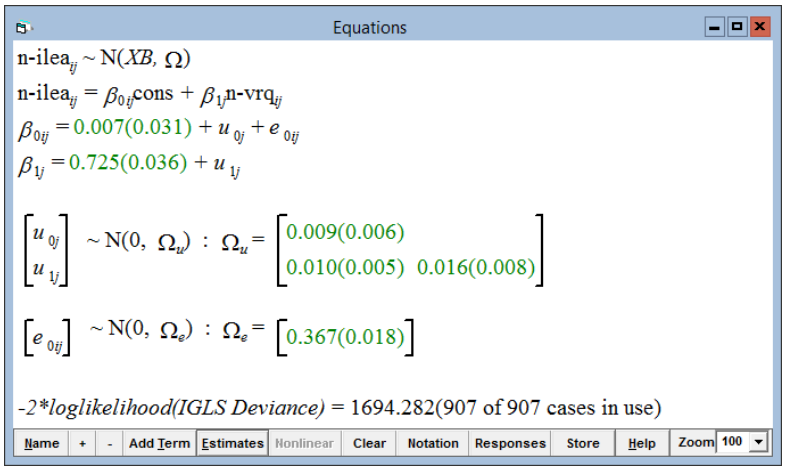
School	10	11	12	13	14	15	16	17	18
# Pupils	47	50	41	49	29	72	62	22	21

The outcome variable for the following analysis is the O-level/CSE examination results, converted into a score by adding up the results for individual subjects for each student using a simple scoring system. The "intake" score for each school is defined as being the Verbal Reasoning quotient, VRQ, a measure of students' ability made when they enter the school. Both of these scores were converted into Normal scores for this analysis, as there was evidence of clustering of scores at high values, probably determined by the fact that there is an upper limit on the scores which an individual student can achieve. Hence our analysis is not exactly equivalent to that of Aitkin and Longford. The outcome variable for each pupil is referred to as **N-ILEA** and the intake variable, measuring VRQ, as **N-VRQ**.

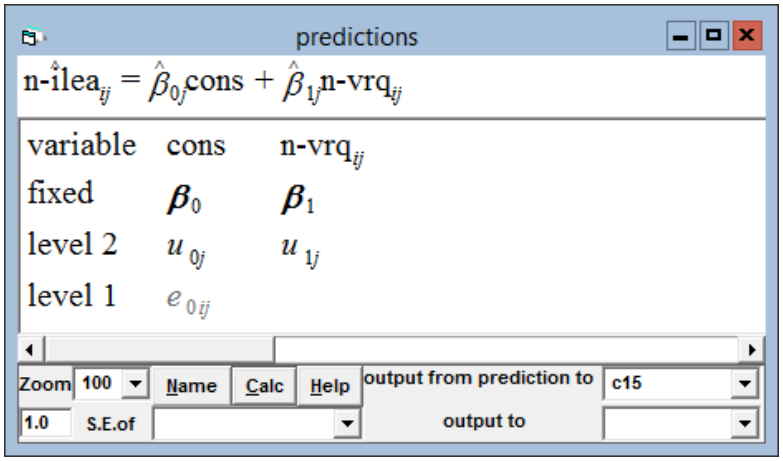
Open the worksheet called **diag1.ws** containing the data, which you can look at using the **Names** and **Data** windows. Go to the **Equations** window to view a model that has already been set up in the worksheet. You will see the following (possibly after clicking on **Names** and / or **Estimates**):


Equations
n-ilea$_{ij}$ ~ N(XB, Ω)
n-ilea$_{ij}$ = β_{0ij}cons + β_{1j}n-vrq$_{ij}$
$\beta_{0ij} = \beta_0 + u_{0j} + e_{0ij}$
$\beta_{1j} = \beta_1 + u_{1j}$
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u01} & \sigma^2_{u1} \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} \sigma^2_{e0} \end{bmatrix}$
-2*loglikelihood(IGLS Deviance) = 1694.282(907 of 907 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

We have a two level model, with students (subscript i) nested within the 18 schools (subscript j). The outcome variable, **N-ILEA**, is modelled as a function of the intake variable, **N-VRQ**, which is also in the random part of the model at level 2, the school level. This means we have a random slopes and intercepts model for schools. There is just one variance term at level 1 measuring the residual variance of the students. Double click on the **Estimates** button at the bottom of the **Equations** window, and run the model until it converges. The result should look like this:

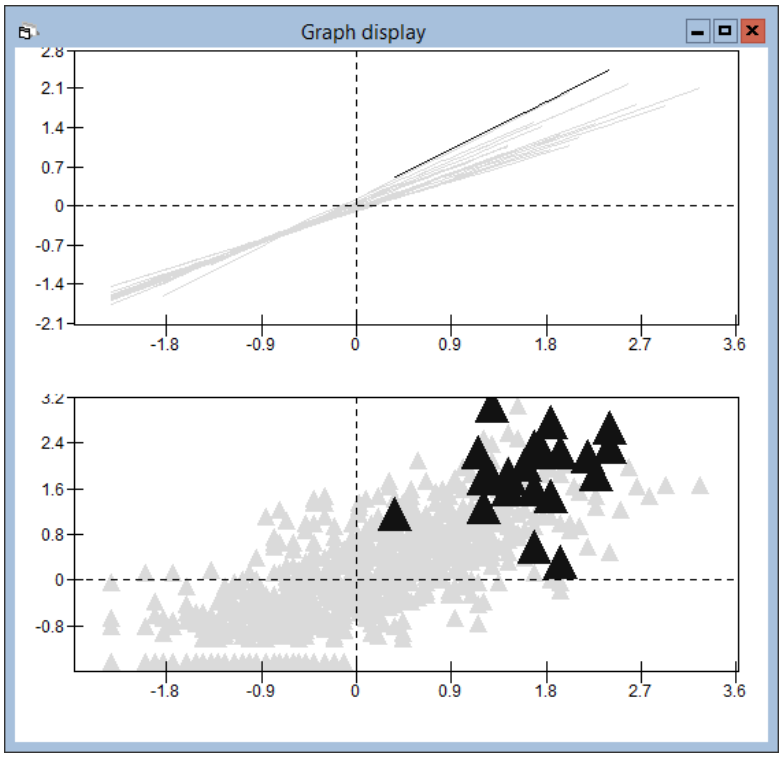

Equations
n-ilea$_{ij}$ ~ N(XB, Ω)
n-ilea$_{ij}$ = β_{0ij}cons + β_{1j}n-vrq$_{ij}$
β_{0ij} = 0.007(0.031) + u_{0j} + e_{0ij}
β_{1j} = 0.725(0.036) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} 0.009(0.006) & \\ 0.010(0.005) & 0.016(0.008) \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} 0.367(0.018) \end{bmatrix}$
-2*loglikelihood(IGLS Deviance) = 1694.282(907 of 907 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

Most of the variance occurs between students, but there is significant variance between schools at level 2. We can explore the relationship between outcome and intake variables for each school by using the **predictions** window. Choose the fixed parameters and random parameters at level 2 to make predictions for the regression lines for each school, and save them into **c15**. The **predictions** window should look like this when you make the calculation:



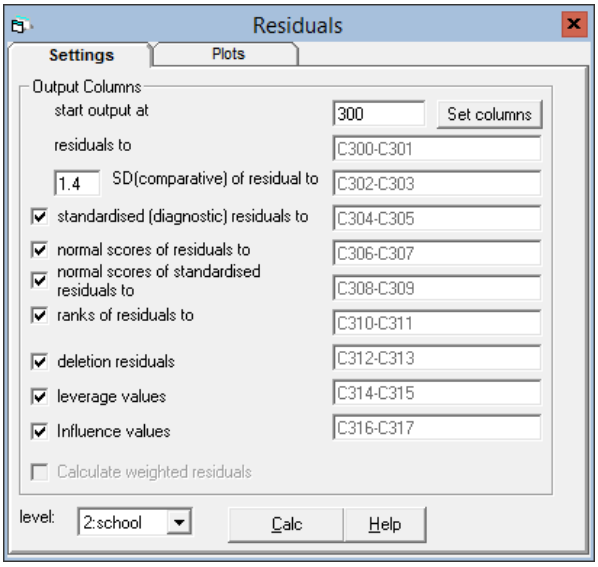
Use the **Customised graph** window to create two graphs in display **D1**. The first is a line graph of the predictions in **c15** plotted against **N-VRQ**, grouped by **SCHOOL**; the second is a point plot of the response variable **N-ILEA** against **N-VRQ**, showing the outcome and intake scores for each student. Highlight the uppermost line on the top graph by clicking just above it. A **Graph options** window will open and inform you that you have identified school 17, one of the grammar schools that will serve as a focus for this example analysis.

From the **In graphs** menu, select **highlight(style 1)** and click **Apply**. The **Graph display** window should look like the following figure, with the school 17 line and the data points for its pupils highlighted in red (shown as dark grey here).



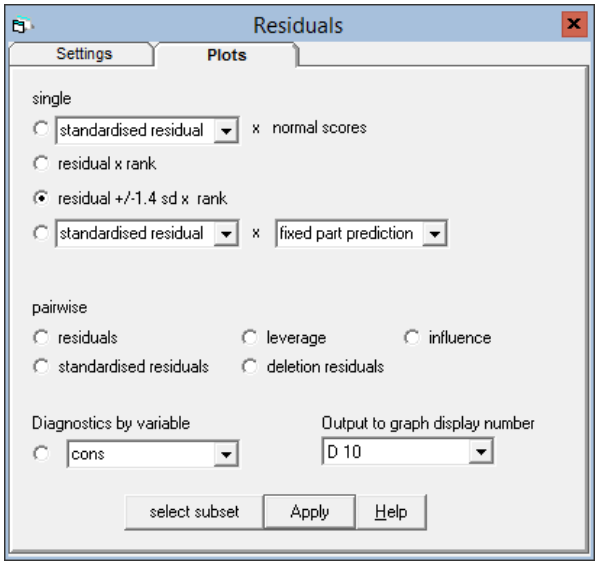
We can see that school 17 has the highest intercept value and one of the highest slope values. From the lower plot, we can also see that students in school 17 tend to be high achievers, with the possible exception of one student whose score is near the mean.

We can examine the values of slopes and intercepts further by using the **Residuals** window. Calculate residuals and other regression diagnostics at level two by selecting **2:SCHOOL** from the **level** drop-down list on the **Settings** tab. Type the value 1.4 into the box beside **SD (comparative) of residual to** so that we can compare confidence intervals around the residuals for each school. The **Residuals** window should now look like the next figure. Remember to click **Calc** to perform the computations.



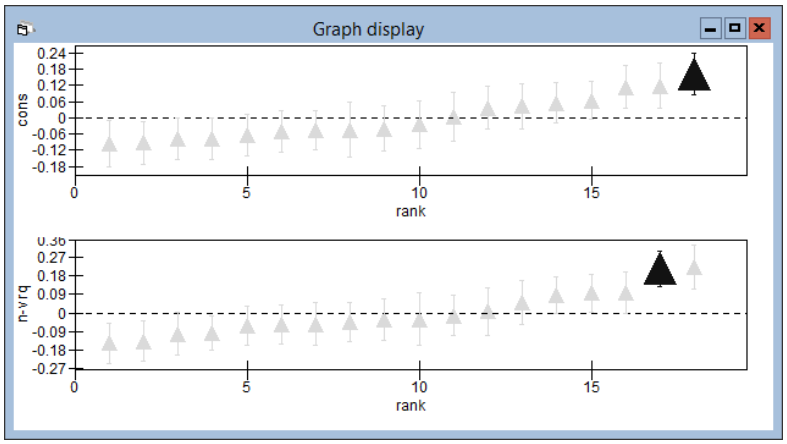
Goldstein & Healy (1995) discuss the circumstances where the value of 1.4 rather than the conventional 1.96 standard deviations is used to calculate 95% intervals. Roughly speaking, if we wish to use the intervals to make comparisons between pairs of schools, then we can judge significance at the 5% level by whether or not the (1.4) intervals overlap. If, on the other hand, we wish, say, to decide whether a school is significantly different from the overall mean, the conventional (1.96) interval can be examined to see whether or not it overlaps the zero line. For present purposes we shall assume that interest focuses on pairwise school comparisons.

Select the **Plots** tab on the **Residuals** window, and then select the option to display **residual +/- 1.4 sd x rank**. The **Residual Plots** menu should look like this:

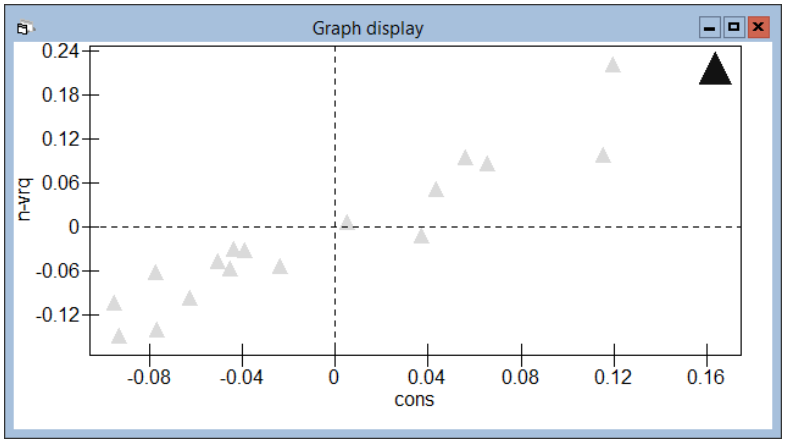


Click **Apply**, and see the following graph displayed.

Note that it is put by default into display **D10**, but you can change this option if you like.



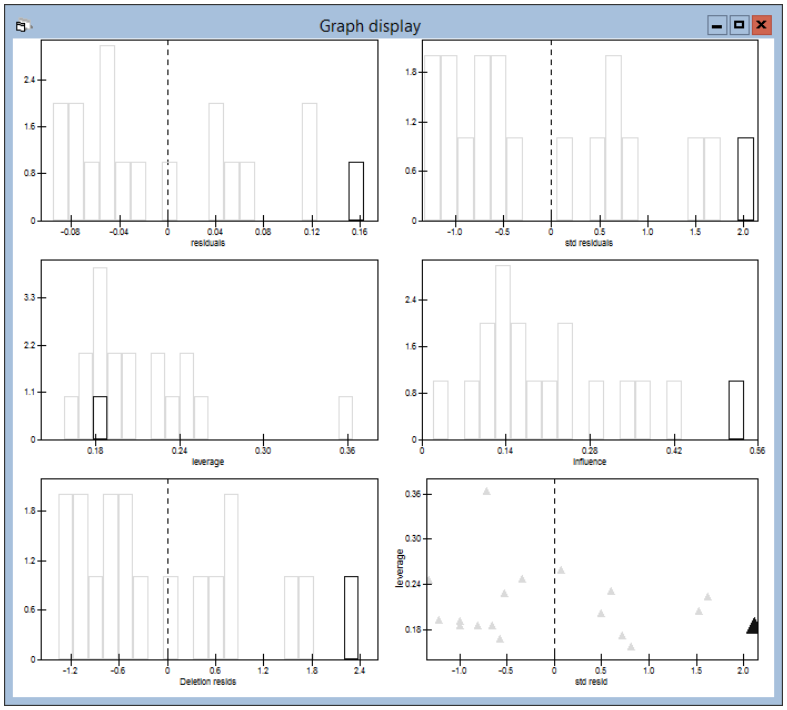
This confirms that school 17 (which we highlighted earlier) has the largest intercept residual, and the second largest slope residual. We can also examine the relationship between intercept and slope residuals. Return to the **Plots** tab of the **Residuals** window, and choose the **residuals** option in the **pairwise** frame, then click on **Apply**. We get this graph as output:



We can see that there is a very strong positive relationship between the values of the intercept and slope residuals, which we can determine from the **Estimates** window is 0.836. This means that the better the average performance of students in a school, the more strongly positive is the relationship between outcome and intake score. School 17 is again shown in the top right hand corner of the graph.

15.2 Diagnostics plotting: Deletion residuals, influence and leverage

We can also examine a number of diagnostic measures at the school level. At the bottom of the **Plots** tab of the **Residuals** window, click on the **diagnostics by variable** box, choosing **CONS** as the variable (this should be shown by default). Click on **Apply**, and the resulting graphics window should look like this:



In the **Graph display** window, we have six plots of diagnostic measures associated with the intercept at the school level, the higher level in the model. School 17, which we have previously chosen to highlight, is shown in red (or dark grey) on all six diagrams. Full explanations of the different diagnostics are given in Langford & Lewis (1998), but a brief descriptive explanation of each measure follows:

1. The plot in the top left hand corner shows a histogram of the raw residuals at school level. As can be seen, school 17 has the highest intercept residual.

2. The top right hand plot is a histogram of the standardised diagnostic residuals at the school level. (See Goldstein (2003) for an explanation of the difference between diagnostic and comparative variances of the residuals). Again, school 17 has the highest standardised residual. The standardised residual (sometimes called the Studentised residual) is the value of the residual divided by its diagnostic standard error, and any value greater than +2 or less than −2 indicates a school which may be significantly different from the mean at the 95% confidence level.

3. The middle left hand plot is a histogram of the leverage values for each school. The leverage values are calculated using the projection, or hat matrix of the fitted values of the model. A high leverage value for a particular school indicates that any change in the intercept for that school will tend to move the regression surface appreciably towards that altered value (Venables & Ripley, 1994). An approximate cut-off point for looking at unusually high leverage values is $2p/n$, where p is the number of random variables at a particular level, and n is the number of units at that level. Here, we have 2 variables and 18 schools,

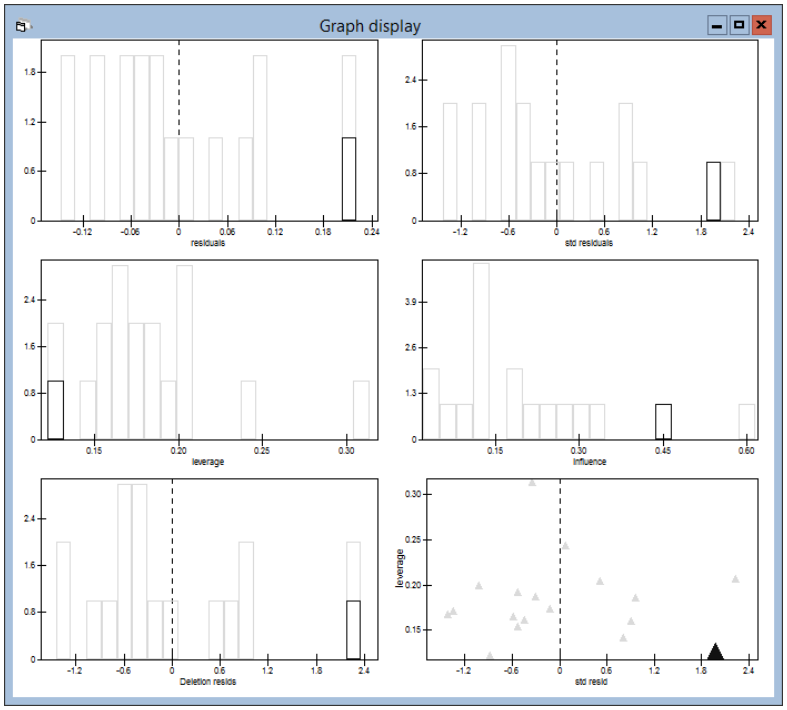
so unusually high leverage values may be above 4/18, i.e. 0.22. One school (6) has a leverage value appreciably above this value, which may require further investigation. School 17 does not have a particularly high leverage value of about 0.17.

4. The middle right hand plot shows influence values, which are a multilevel equivalent of the DFITS measure of influence (Belsey et al. (1980); Velleman & Welsch (1981)). The influence values combine the residuals and leverage values to measure the impact of each school on the coefficient value for, in this case, the intercept (see Langford & Lewis (1998) for further details). School 17, having a large residual, and a roughly average leverage value comes out as having the highest influence on the intercept.

5. The lower left hand plot is a histogram of deletion residuals. The deletion residuals show the deviation between the intercept for each particular school and the mean intercept for all schools, when the model is fitted to the data excluding that school. When the number of units at a particular level is large, these will be very similar to the standardised residuals. However, when the number of schools is small — in this case, there are only 18 schools — there may be some differences. It is the deletion residuals that are used in the calculation of influence values discussed above.

6. The lower right hand diagram shows a plot of leverage values against standardised residuals. We can see school 17 at the far right of the graph, and school 6, with a high leverage value is towards the top left hand corner. Schools with unusual leverage values or residuals can easily be identified from the plot using the mouse.

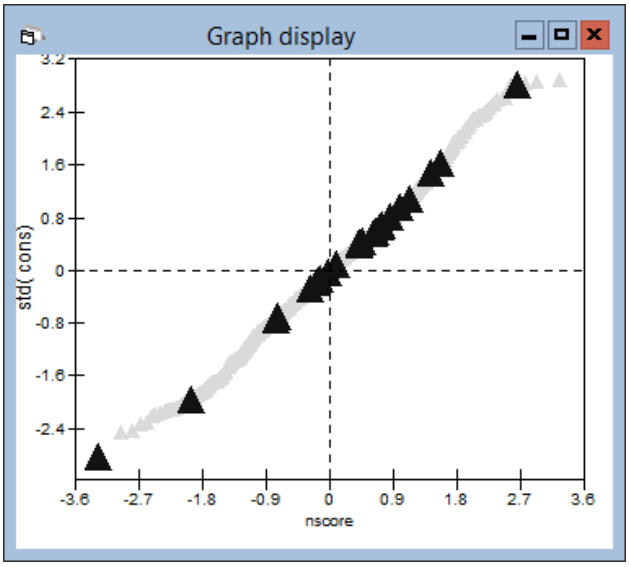
We can calculate the same measures for the slopes at the school level associated with the explanatory variable **N-VRQ**. If you return to the **Residuals** window and this time choose **N-VRQ** in the **diagnostics by variable** box and click **Apply**, the **Graph display** window should look like this:



We can see from the standardised residuals plot that two schools have unusually high slope values, these being school 17 and school 12. School 12 also has the highest influence on the slope, followed by school 17, so school 12 would be a good candidate for further investigation, although for the sake of simplicity, we shall focus our attention on school 17 in this analysis.

Return to the **Settings** tab of the **Residuals** window, and choose **1:pupil** for **level**. Type **320** in the box beside the **Set columns** button, and click on the button to store the computed values in a group of columns beginning at **c320**. This way, we retain the columns we have used for the school-level diagnostics.

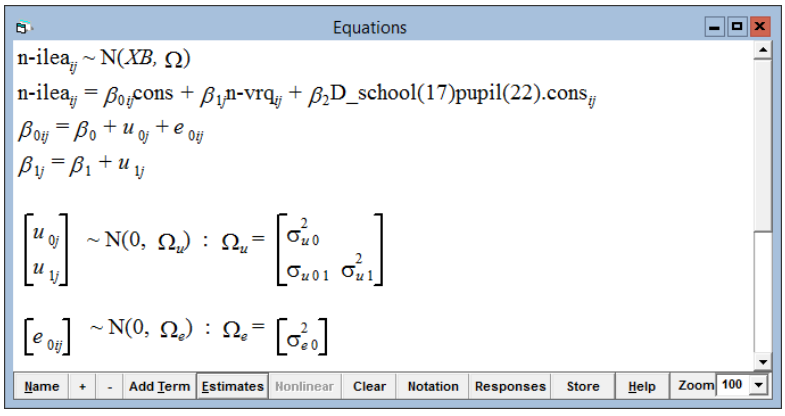
Now select the **Plots** tab of the **Residuals** window, and choose to plot standardised residuals against Normal scores. Note that we only have one set of residuals at level 1, as only **CONS** is randomly varying at pupil level. The graph window should look like this:



The pupils in School 17 are shown as red (or dark grey) triangles. As we can see, there is one pupil at the bottom left of the plot who has a particularly high negative residual, i.e. he is a low achiever in the overall high achieving school 17.

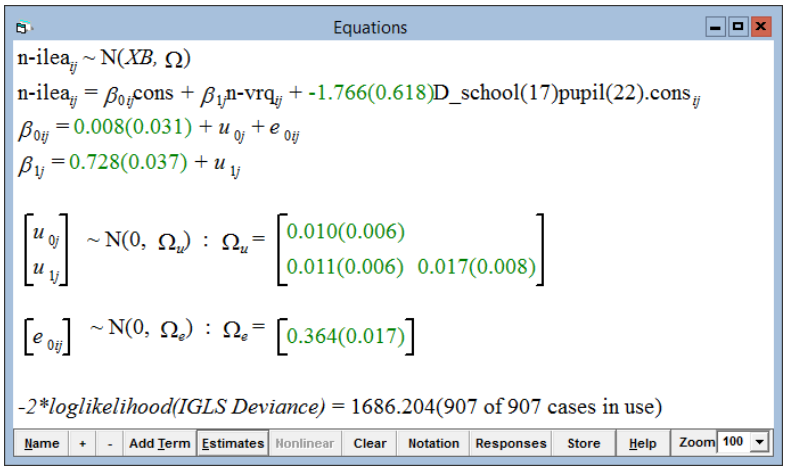
We will now examine the effect on the model of omitting this low achiever from the random effects for school 17 by fitting a separate value for the low achiever in the fixed part of the model. Click the left mouse button while pointing at the red triangle for this pupil, which will pick the individual out as pupil 22 in school 17. From the **In graphs** menu on the **Identify point** tab of the **Graph options** window, choose to highlight this pupil with **Highlight(style 2).** (Pick out the option and then click on **Apply**). Next, choose the **Absorb into dummy** option from the **In model** box, and click on **Apply**. If you return to the **Graph display** window, this particular pupil should now be shown as a light blue triangle.

Now open the **Equations** window, which will have been updated to look this:



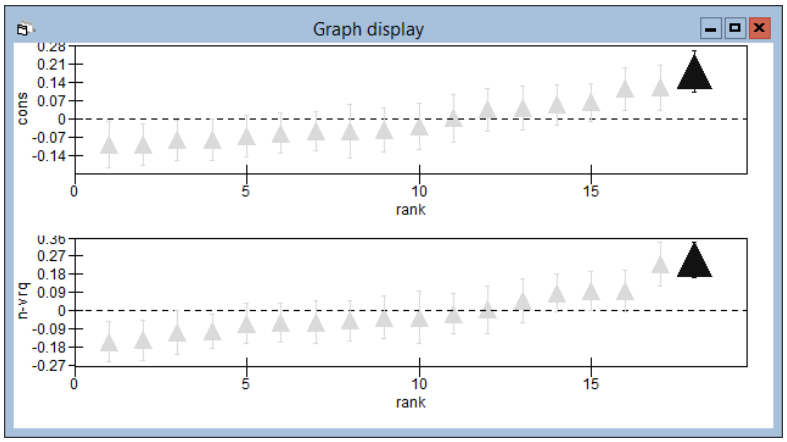
The dummy variable for school 17, pupil 22 has been added to the fixed part of model, thereby excluding this pupil from the random part of the model at

the pupil level. To update the parameter estimates, click on **More**, and you should get this result:


Equations
n-ilea$_{ij}$ ~ N(XB, Ω)
n-ilea$_{ij}$ = β_{0ij}cons + β_{1j}n-vrq$_{ij}$ + -1.766(0.618)D_school(17)pupil(22).cons$_{ij}$
β_{0ij} = 0.008(0.031) + u_{0j} + e_{0ij}
β_{1j} = 0.728(0.037) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} 0.010(0.006) & \\ 0.011(0.006) & 0.017(0.008) \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix}$ ~ N(0, Ω_e) : Ω_e = $\begin{bmatrix} 0.364(0.017) \end{bmatrix}$
-2*loglikelihood(IGLS Deviance) = 1686.204(907 of 907 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

Note that the residual deviance has dropped by a little over 8 units for the inclusion of this one extra fixed parameter, an improvement in the model that is significant at the 0.01 level.

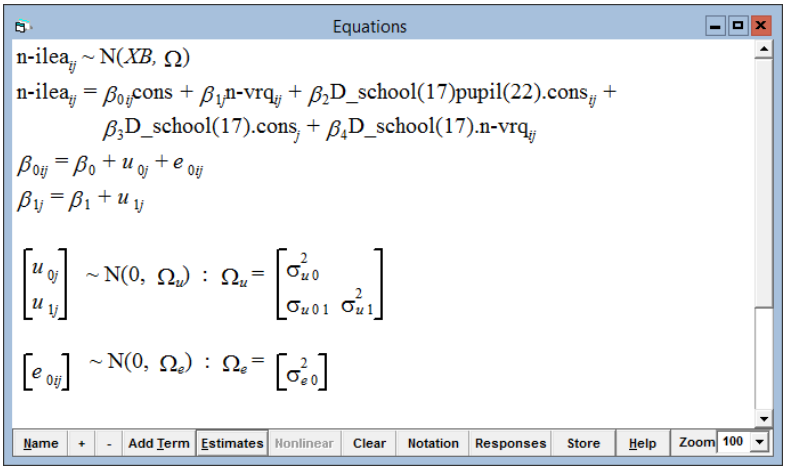
We can now re-examine the school-level residuals obtained from the new model. Return to the **Settings** tab of the **Residuals** window, and choose to calculate school-level residuals in columns **c330** onwards. Go to the **Plots** tab and choose to plot **residuals +/- 1.4 sd x rank** again. The graph window looks like this:



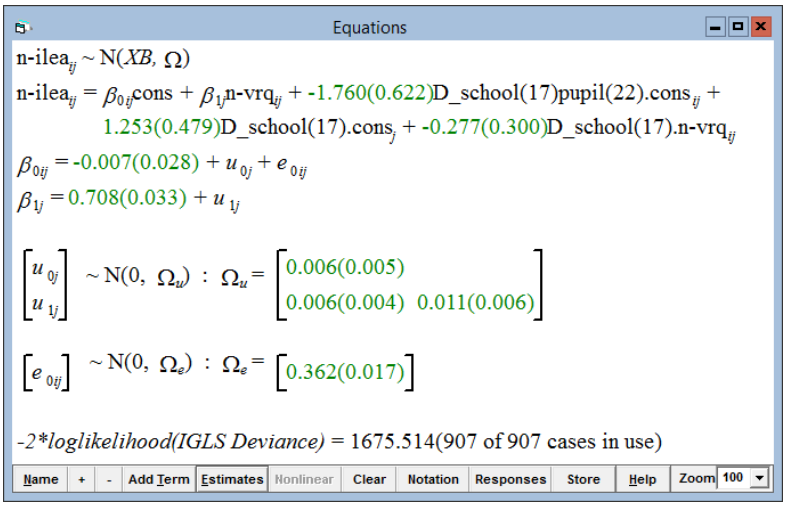
We now see that omitting the low achieving pupil *within* school 17 has made school 17 even more extreme than previously for both intercept and slope at the school level.

We can choose to create dummy variables for school 17, to fit an intercept and slope separately from those of the other schools. Begin by selecting one of the school 17 points on the caterpillar plot and using the **Identify point** tab of the **Graph options** window, as before. Choose the **Absorb into dummy** variable from the **In model** box again. This time when you click on **Apply**, another window will open, asking whether you want to fit

terms for interactions with **CONS** and/or **N-VRQ**. Make sure both variables are selected, and click on **Done**. The **Equations** window is updated automatically, and should now look like this:

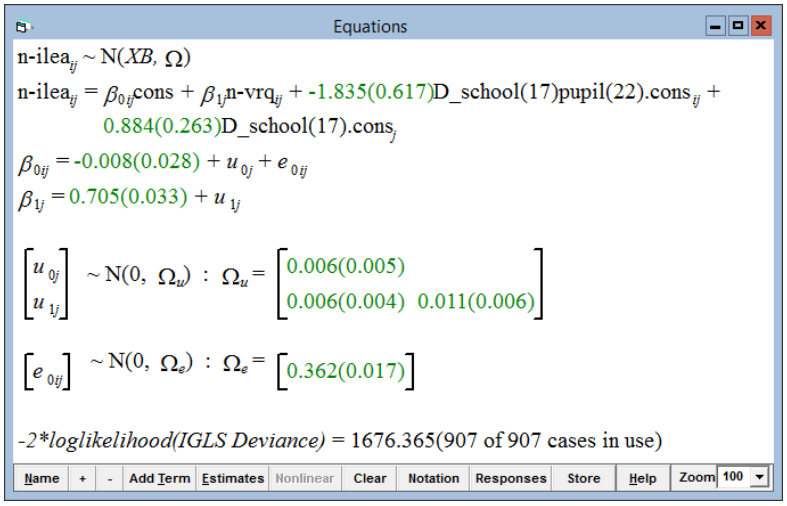

Equations
n-ilea$_{ij}$ ~ N(XB, Ω)
n-ilea$_{ij}$ = β_{0ij}cons + β_{1j}n-vrq$_{ij}$ + β_2D_school(17)pupil(22).cons$_{ij}$ +
β_3D_school(17).cons$_j$ + β_4D_school(17).n-vrq$_{ij}$
β_{0ij} = β_0 + u_{0j} + e_{0ij}
β_{1j} = β_1 + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω$_u$) : Ω$_u$ = $\begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u01} & \sigma^2_{u1} \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix}$ ~ N(0, Ω$_e$) : Ω$_e$ = $\begin{bmatrix} \sigma^2_{e0} \end{bmatrix}$
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

We can see that two extra variables have been added into the model: an intercept term for school 17, and a slope term. Click on **More** and wait for the model to converge. The result should be:


Equations
n-ilea$_{ij}$ ~ N(XB, Ω)
n-ilea$_{ij}$ = β_{0ij}cons + β_{1j}n-vrq$_{ij}$ + -1.760(0.622)D_school(17)pupil(22).cons$_{ij}$ +
1.253(0.479)D_school(17).cons$_j$ + -0.277(0.300)D_school(17).n-vrq$_{ij}$
β_{0ij} = -0.007(0.028) + u_{0j} + e_{0ij}
β_{1j} = 0.708(0.033) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω$_u$) : Ω$_u$ = $\begin{bmatrix} 0.006(0.005) & \\ 0.006(0.004) & 0.011(0.006) \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix}$ ~ N(0, Ω$_e$) : Ω$_e$ = $\begin{bmatrix} 0.362(0.017) \end{bmatrix}$
-2*loglikelihood(IGLS Deviance) = 1675.514(907 of 907 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

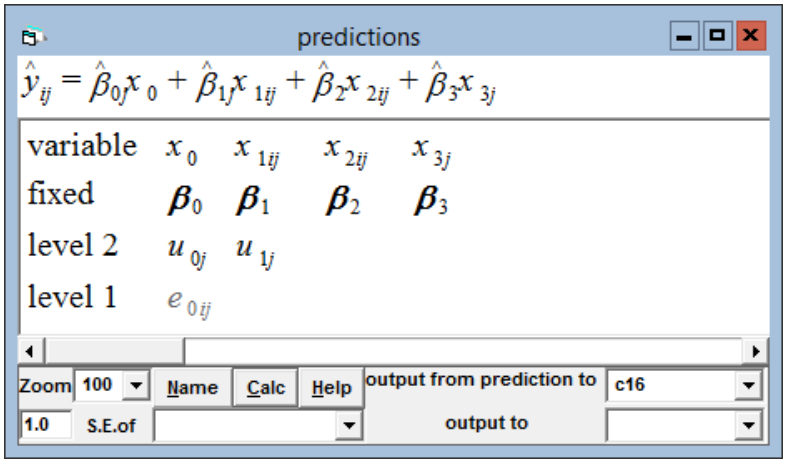
Note that the random parameter estimates at school level for the variances of slopes and intercepts have both dropped noticeably, now that the most extreme school has been excluded from the random part of the model. The overall decrease in residual deviance is about 10.7, with the loss of two degrees of freedom for the extra parameters included. This is highly significant, but if we look at the individual parameter estimates, we can see that the intercept value for school 17 is highly significant, whilst the slope value is not.

We can examine the effect of excluding the slope-related variable for school 17 by clicking on **D_SCHOOL(17).N-VRQ** in the **Equations** window, and choosing to **Delete term** from the model. Refit the model using **More**, and the results look like the following:

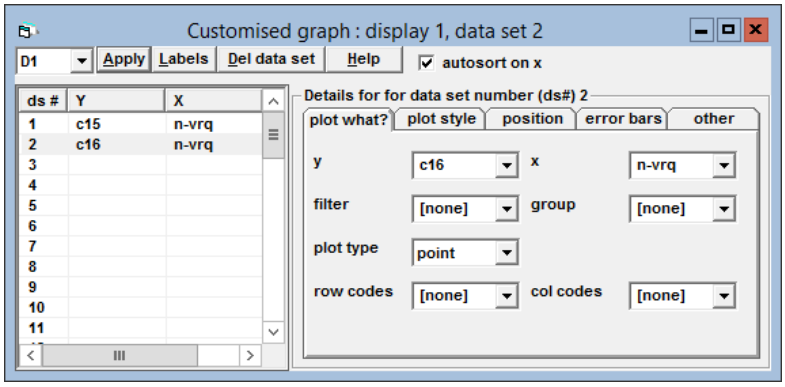

Equations
n-ilea$_{ij}$ ~ N(XB, Ω)
n-ilea$_{ij}$ = β_{0ij}cons + β_{1j}n-vrq$_{ij}$ + -1.835(0.617)D_school(17)pupil(22).cons$_{ij}$ +
0.884(0.263)D_school(17).cons$_j$
β_{0ij} = -0.008(0.028) + u_{0j} + e_{0ij}
β_{1j} = 0.705(0.033) + u_{1j}
$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix}$ ~ N(0, Ω$_u$) : Ω$_u$ = $\begin{bmatrix} 0.006(0.005) & \\ 0.006(0.004) & 0.011(0.006) \end{bmatrix}$
$\begin{bmatrix} e_{0ij} \end{bmatrix}$ ~ N(0, Ω$_e$) : Ω$_e$ = $\begin{bmatrix} 0.362(0.017) \end{bmatrix}$
-2*loglikelihood(IGLS Deviance) = 1676.365(907 of 907 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store Help Zoom 100

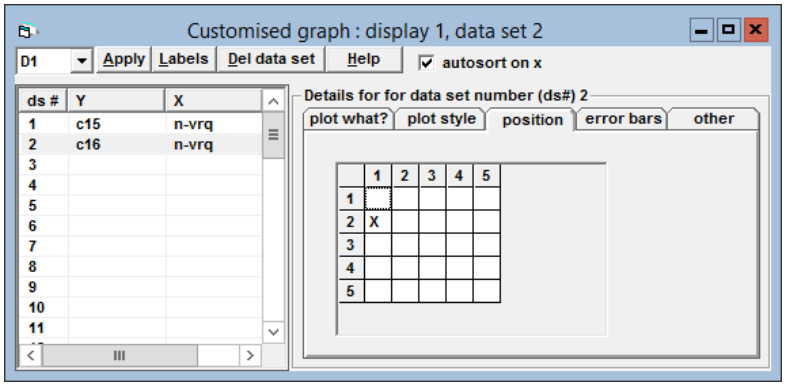
The residual deviance has only changed by a small amount, so we conclude that greater parsimony in our model is achieved by fitting a separate intercept for school 17 and a separate fixed effect for pupil 22 in school 17.

We can examine the predictions for the school by using the **predictions** window as before. You should set it up to look like the following (saving the predictions into **c16**) before clicking on **Calc**:

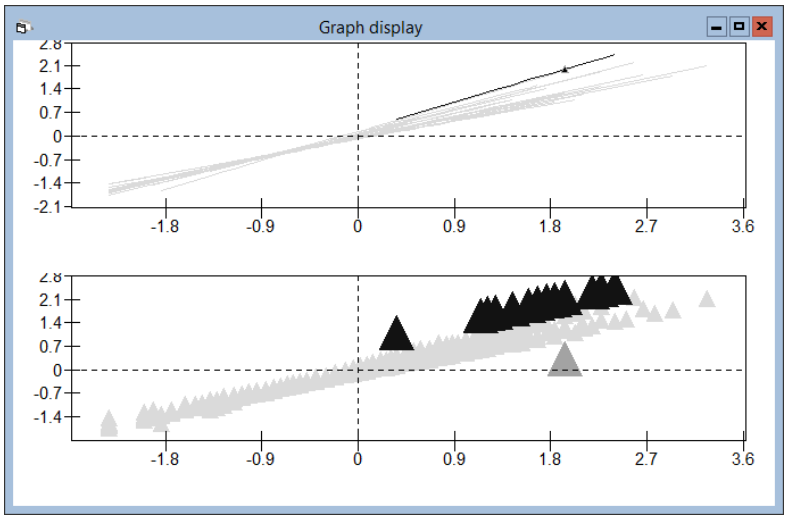

predictions
$\hat{y}_{ij} = \hat{\beta}_{0j}x_0 + \hat{\beta}_{1j}x_{1ij} + \hat{\beta}_2x_{2ij} + \hat{\beta}_3x_{3j}$
variable x_0 x_{1ij} x_{2ij} x_{3j}
fixed β_0 β_1 β_2 β_3
level 2 u_{0j} u_{1j}
level 1 e_{0ij}
Zoom 100 Name Calc Help output from prediction to c16
1.0 S.E.of output to

Now go to the **Customised graph** window, and choose display **D1** as before. Leave data set 1 as it was, and choose to edit data set 2 to show the new predictions from the current model. Before you click on **Apply**, the **plot what?** and **position** tabs should look this:





Note that we are choosing to plot points, rather than lines. The resulting graph should look as below, showing at the top our initial predictions from the model as a line graph, with school 17 highlighted, and at the bottom, our new predictions, with pupils in school 17 highlighted as red (or dark grey) triangles. Note the pupil (22) who does not fit on the school's line, i.e., the one we have chosen to exclude from the random part of the model. Note also that there is another low achieving pupil that we could choose to exclude from the model (see Langford & Lewis (1998) for more details on this).



We can then explore the regression diagnostics, choosing to examine particular schools or pupils as before in more detail.

15.3 A general approach to data exploration

Suppose we are particularly interested in observations that are outlying with respect to any of the terms with random coefficients in a model. Where should we start data exploration after fitting a multilevel model? Rather than looking at individual data points, we have found it most useful to begin at the level of highest aggregation, which will often be simply the highest level in the model. There are two reasons for this. Researchers are often most interested in the highest level of aggregation, and will naturally concentrate their initial efforts here. However, if discrepancies can be found in higher-level structures, these are more likely to be indicative of serious problems than a few outlying points in lower-level units.

After the highest level has been analysed, lower levels should be examined in turn, with analysis and initial treatment of outliers at the lowest level of the model. The highest level should then be re-examined after a revised model has been fitted to the data. The objective is to determine whether an outlying unit at a higher level is entirely outlying, or outlying due to the effects of one or two aberrant lower-level units it contains.

Similarly, examination of lower-level units may show that one or two lower-level units are aberrant *within* a particular higher-level unit that does not appear unusual, and that the higher-level unit would be aberrant without these lower-level units. Hence, care must be taken with the analysis not simply to focus on interesting higher-level units, but to explore fully lower-level units as well.

Chapter learning outcomes

- ⋆ The use of diagnostic procedures for exploring multilevel models
- ⋆ The importance of studying data carefully to check model assumptions
- ⋆ How to deal with 'discrepant' measurements

Chapter 16

An Introduction to Simulation Methods of Estimation

In the previous chapters we have used a variety of estimation procedures: IGLS, RIGLS and MQL for Normal responses, and MQL and PQL for discrete responses. These estimation procedures are all deterministic in that, given a data set and a model, they always converge in the same number of iterations to the same estimates. If you run the estimation procedure 100 times you get the same answers every time.

Estimation procedures can also be *stochastic*, that is they contain within them simulation steps in which random numbers are sampled. These simulation steps mean that every time you run the estimation procedure you get a slightly different estimate. This obviously raises the question of what is the correct estimate? Although this uncertainty may appear to be a disadvantage, simulation methods often have important advantages. For example, more accurate, less biased estimates are delivered, and complex models are more easily accommodated.

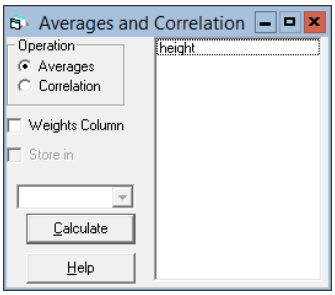
There are two families of simulation based estimation procedures available in MLwiN: MCMC sampling and bootstrapping. The MCMC sampling facilities in MLwiN are very extensive and are described in Browne (2003).

This chapter in introduces the reader to some basic simulation ideas, and Chapter 17 describes the bootstrap facilities available in MLwiN.

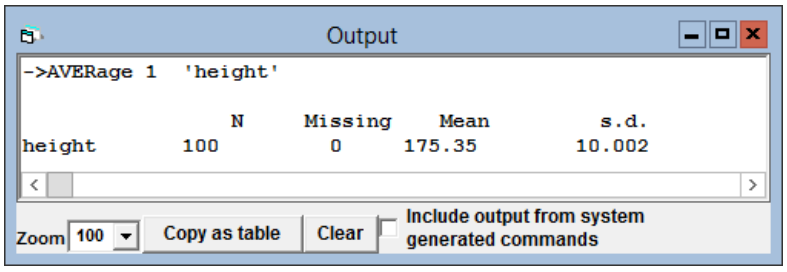
16.1 An illustration of parameter estimation with Normally distributed data

Probably the most commonly studied data in the field of statistics involve variables that are continuous, and the most common distributional assumption for continuous data is that they are Normally distributed. One example of a continuous data set that we will consider here is the heights of adult males. If you were asked to estimate the average height of adult males in Britain, how would you provide a good estimate? One approach would be to take a simple random sample, that is travel around the country measuring a sample of the population. Then from this sample we could calculate the mean and use it as an estimate.

The worksheet **height.ws** has one column of data, named **Height**, which contains the heights of 100 adult males measured in centimetres. Open this worksheet using the **Open Worksheet** option on the **File** menu. You can calculate the average height of the sample members via the **Averages and Correlation** window that can be accessed from the **Basic Statistics** menu:



Select **Height** from the column list on the right of the window and click on **Calculate**. The Output window will appear and the following results are given for our sample:

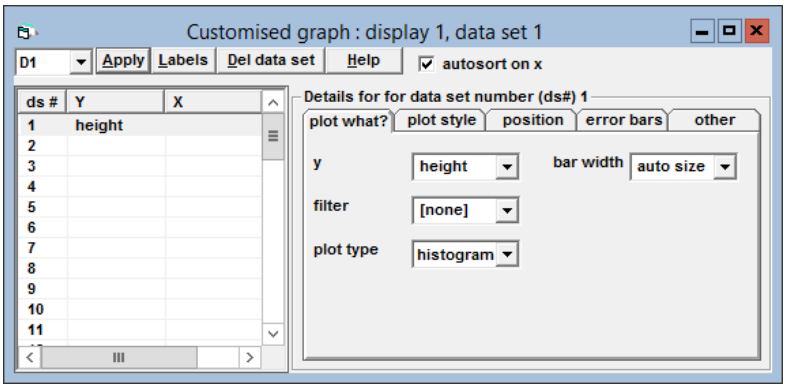


Obviously the larger the sample of people that are measured, the more accurate the mean estimate will be, and consequently the better estimate the sample mean will be for the population mean. We can also plot the 100 heights as a histogram to give a graphical description of this data set.

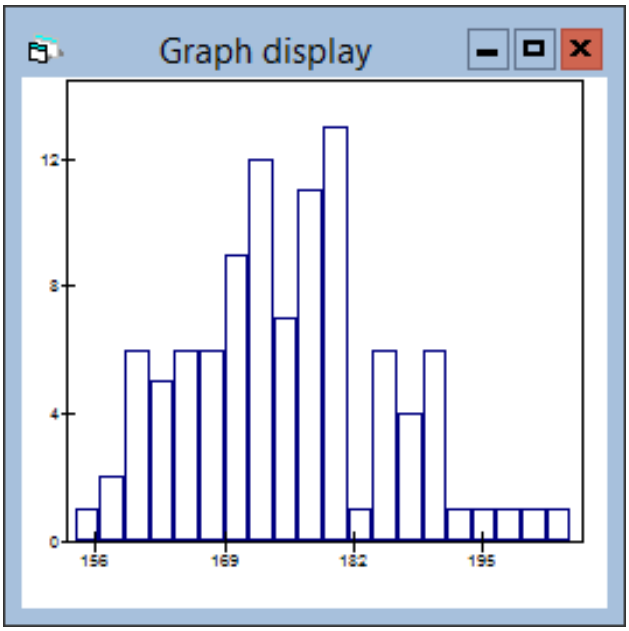
- Select the **Customised graph** window from the **Graphs** menu

(shown below)

- Select **Height** from the **y** list
- Select **histogram** from the **plot type** list
- Click **Apply**



The histogram shown below will now appear. Note that the shading pattern for the bars and their colour can be altered with the options provided on the **plot style** tab.



Another question that could be asked is 'What percentage of British adult males are over 2 metres tall?' We can consider two approaches to answering this question, a *non-parametric* approach or a *parametric* approach. The non-parametric approach would be to calculate directly from the list of 100 heights the percentage of heights that are above 2 metres. In this case the percentage is 1% as only 1 height is greater than 2 metres. The parametric approach would involve making an assumption about the distribution of the data. Typically we would assume that the data are Normally distributed, which from the histogram of the 100 heights looks feasible.

We would then need to find the probability of getting the value 200 from a Normal distribution with a mean of 175.35 and a standard deviation of 10.002. MLwiN provides tail probabilities for a standard Normal distribution, so we will need to Z transform our value to give $(200 - 175.35)/10.002 = 2.4645$. Then we use the **Tail Areas** window from the **Basic Statistics** menu as follows:

- Select **Standard Normal distribution** from the **Operation** list
- Type **2.4645** in the **Value** box
- Click on **Calculate**

This gives the following value in the **Output** window:

```
NPRObability 2.4645 0.0068602
```

The parametric approach estimates that 0.68% of the population is over 2 metres tall. We will be considering parametric and non-parametric approaches again when we discuss bootstrap simulation methods.

We have not as yet used any simulation-based methods. We will now consider the problem of making inferences about the mean and variance of the population from which our sample of 100 males has been taken. Here we will discuss three methods.

Normal Distribution Sampling Theory

In the case of a sample of Normally distributed observations of size N, distributions of the sample mean, $\bar{x}$, and variance, s^2, can be calculated from sampling theory. The sample mean is Normally distributed with mean μ and variance σ^2/N. Consequently a 95% central confidence interval for μ is $\bar{x} \pm 1.96\sigma/\sqrt{N}$. In our sample of men's heights, a 95% central confidence interval for μ is $175.35 \pm 1.96 \times 10.002/\sqrt{100} = (173.39, 177.31)$.

The population variance, σ^2 is related to the sample variance, s^2 by the Chi-squared distribution as follows: $(N-1)s^2/\sigma^2 \sim \chi^2_{N-1}$. Consequently a 95% central confidence interval for σ^2 is $((N-1)s^2/\chi^2_{(N-1),0.025}, (N-1)s^2/\chi^2_{(N-1),0.975})$. In our sample a 95% central confidence interval for σ^2 is (77.12,135.00). Note that this interval is not symmetric.

Parametric and Nonparametric Bootstrapping

The mean and variance for a single sample gives us one estimate for each population parameter. If we could get a *sample* of mean estimates and a

sample of variance estimates then we could use these samples to construct interval estimates for the underlying parameters. This idea of generating a large number of samples to create interval estimates is the motivation behind most simulation methods.

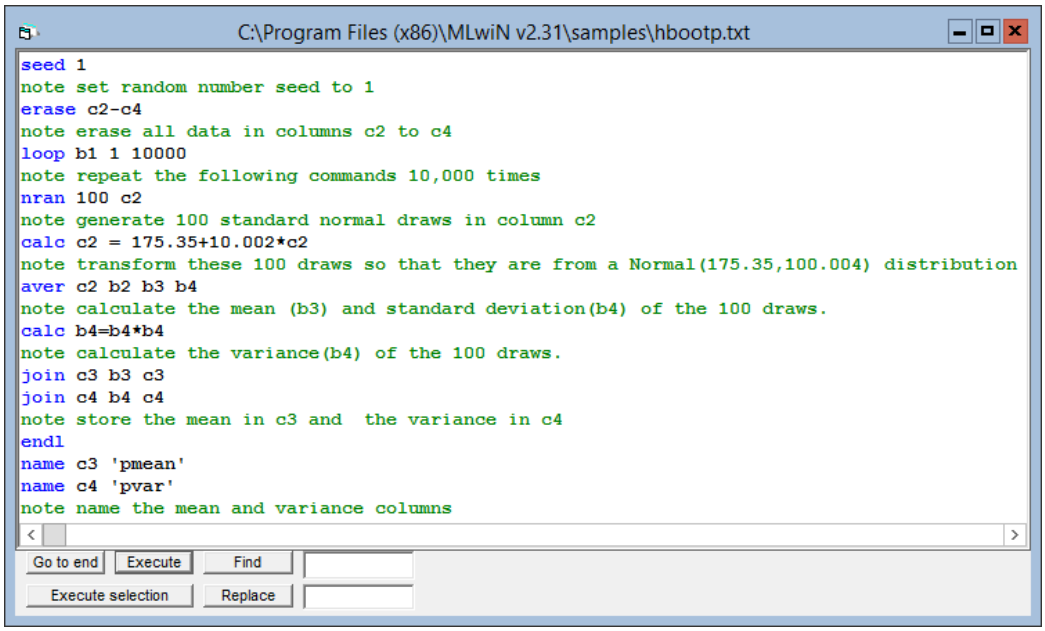
Bootstrapping works by constructing a series of data sets similar to our actual data set (using the actual data set as an estimate of the population distribution) and then using these data sets to summarise the parameters of interest. The way the data sets are constructed depends on which type of bootstrapping is used.

Parametric bootstrapping uses assumptions about the distribution of the data to construct the bootstrap data sets. Consider our sample of 100 heights that has a mean of 175.35 and a standard deviation of 10.002. To create parametric bootstrap data sets from this data set, we simply draw multiple samples of size 100 from a Normal(175.35, $(10.002)^2$) distribution. Then for each sample we calculate the parameter we are interested in.

To illustrate this bootstrapping procedure in MLwiN we will introduce the MLwiN macro language. Using macros makes it simple to run a series of commands[1] in MLwiN repeatedly. We will now use a simple macro to perform parametric bootstrap estimation using our sample of 100 heights.

- Select **Open Macro** from the **File** menu
- From the list of files, select **hbootp.txt** and click **Open**

The **macro** window shown below should now appear.


C:\Program Files (x86)\MLwiN v2.31\samples\hbootp.txt
seed 1
note set random number seed to 1
erase c2-c4
note erase all data in columns c2 to c4
loop b1 1 10000
note repeat the following commands 10,000 times
nran 100 c2
note generate 100 standard normal draws in column c2
calc c2 = 175.35+10.002*c2
note transform these 100 draws so that they are from a Normal(175.35,100.004) distribution
aver c2 b2 b3 b4
note calculate the mean (b3) and standard deviation(b4) of the 100 draws.
calc b4=b4*b4
note calculate the variance(b4) of the 100 draws.
join c3 b3 c3
join c4 b4 c4
note store the mean in c3 and the variance in c4
endl
name c3 'pmean'
name c4 'pvar'
note name the mean and variance columns
Go to end
Execute
Find
Execute selection
Replace

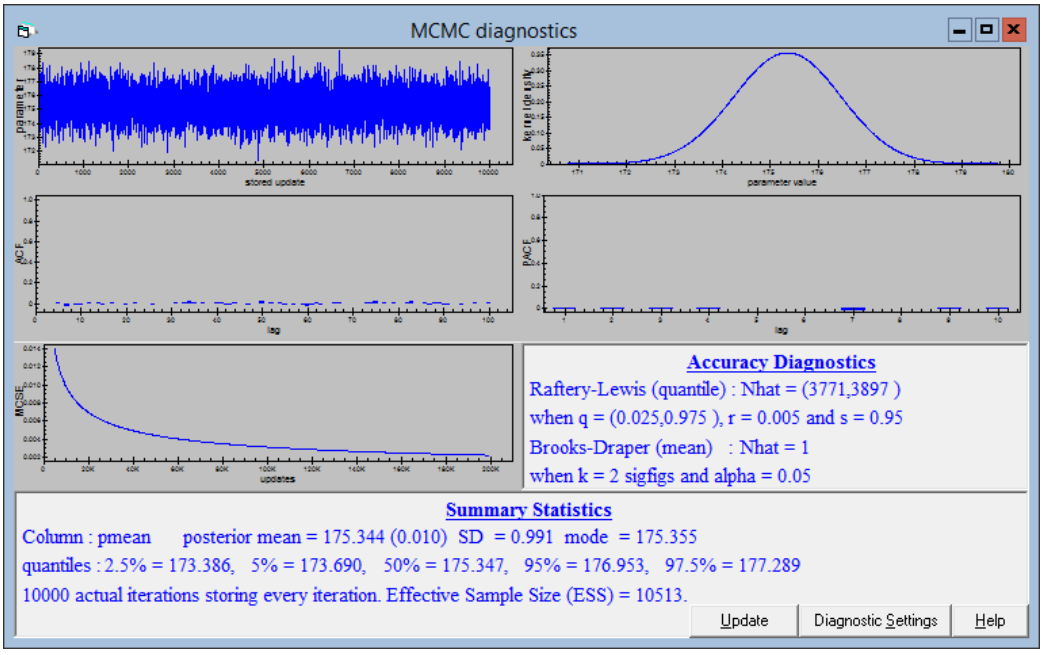
[1]Virtually all menu operations in MLwiN have an equivalent command that can be typed into the **Command interface** window or executed via a macro. These commands are documented in MLwiN's on-line **Help** system under **Commands**.

This macro above is designed to generate 10,000 samples of size 100 from a Normal distribution with mean 175.35 and standard deviation 10.002. Then for each sample, the mean is stored in column **c3** named **pmean**, and the variance is stored in column **c4** named **pvar**.

The MLwiN commands are highlighted in blue. Highlighting a command in the macro window and pressing F1 will bring up the **Help** documentation for that command. The **note** command, on the green lines, is used to add explanatory comments to a macro. All lines starting with **note** are ignored when the macro is run.

Now click on **Execute** to run the macro. After a short time the mouse pointer will change back from an hourglass symbol to a pointer to signify that the macro has finished. We would now like to look at the chains of mean and variance values in more detail.

Open the **Column Diagnostics** window from the **Basic Statistics** menu and select the column labelled **pmean**. Now click the **Apply** button, and after a short wait the **MCMC diagnostics** window will appear as below. This diagnostics window is generally used for MCMC chains as described in the 'MCMC Estimation in MLwiN' manual, so a lot of the information on the window is irrelevant at this point.

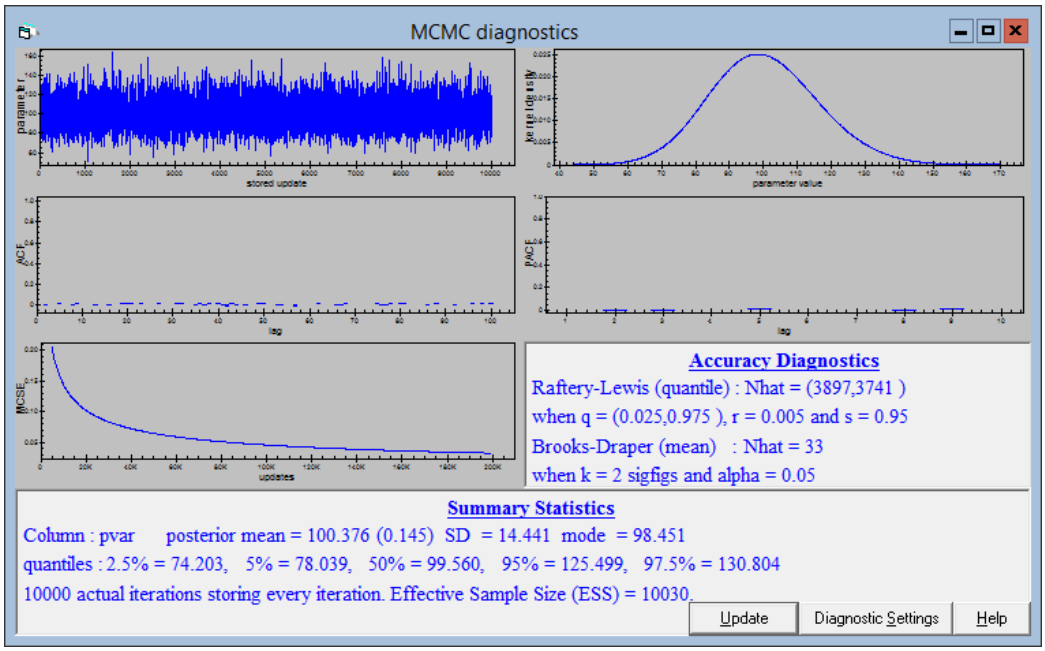

MCMC diagnostics
Accuracy Diagnostics
Raftery-Lewis (quantile) : Nhat = (3771,3897)
when q = (0.025,0.975), r = 0.005 and s = 0.95
Brooks-Draper (mean) : Nhat = 1
when k = 2 sigfigs and alpha = 0.05
Summary Statistics
Column : pmean posterior mean = 175.344 (0.010) SD = 0.991 mode = 175.355
quantiles : 2.5% = 173.386, 5% = 173.690, 50% = 175.347, 95% = 176.953, 97.5% = 177.289
10000 actual iterations storing every iteration. Effective Sample Size (ESS) = 10513.
Update
Diagnostic Settings
Help

We will concentrate on the top two graphs and the summary statistics box. The graph on the left is a trace plot of the means of the 10,000 samples in the order that they were generated. These 10,000 values have been used to construct the kernel density graph on the right. A kernel density plot (Silverman, 1986) is like a smoothed version of a histogram. Instead of being allocated to an appropriate histogram bin, each value's contribution to the graph is allocated smoothly via a kernel function. As we can see, the mean

appears to be Normally distributed, which is what we expected.

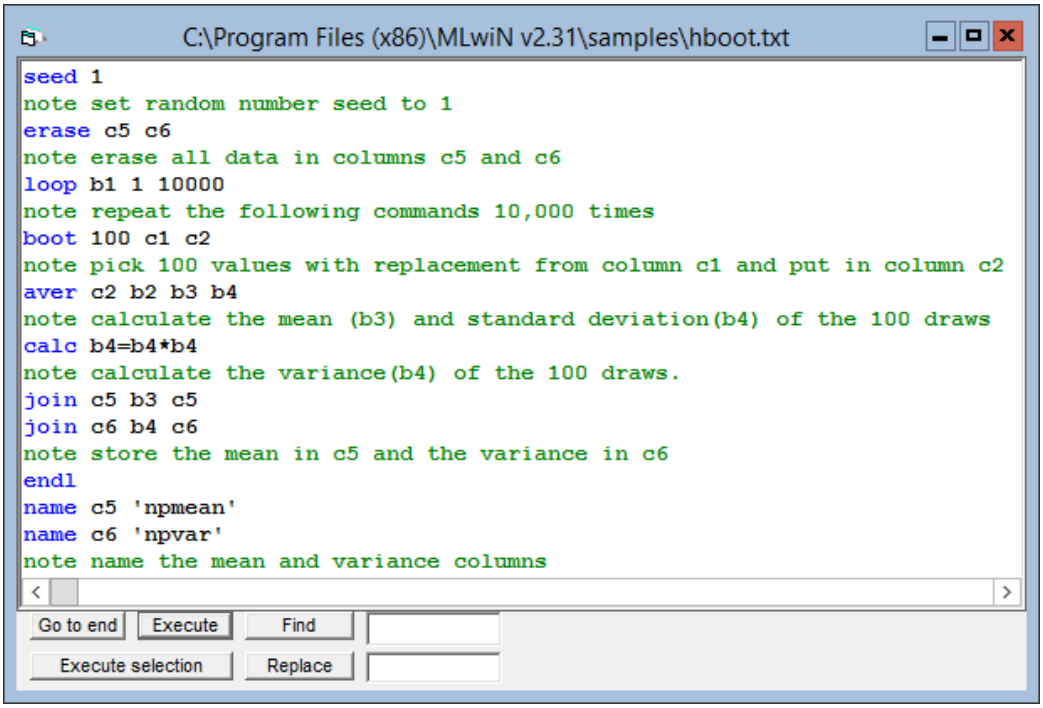
The bottom box contains various summary statistics for the mean parameter, including quantiles that can be used to construct an interval estimate. Here a 95% central interval is (173.39, 177.29) which compares favourably as it should with the theoretical interval of (173.39, 177.31).

We will also look at the population variance. Select the variable **pvar** from the **Column Diagnostics** window and click **Apply**. The MCMC diagnostics window for the variance parameter will appear.


MCMC diagnostics
parameter
stored update
kernel density
parameter value
ACF
lag
PACF
lag
MCSE
updates
Accuracy Diagnostics
Raftery-Lewis (quantile) : Nhat = (3897,3741)
when q = (0.025,0.975), r = 0.005 and s = 0.95
Brooks-Draper (mean) : Nhat = 33
when k = 2 sigfigs and alpha = 0.05
Summary Statistics
Column : pvar posterior mean = 100.376 (0.145) SD = 14.441 mode = 98.451
quantiles : 2.5% = 74.203, 5% = 78.039, 50% = 99.560, 95% = 125.499, 97.5% = 130.804
10000 actual iterations storing every iteration. Effective Sample Size (ESS) = 10030
Update
Diagnostic Settings
Help

Here we see that the kernel density plot does not now look quite like a Normal distribution and has a slightly longer right hand tail. This was also to be expected based on the theoretical distribution of the variance. Now if we look at the confidence interval constructed by the quantiles we get (74.203,130.804) which is similar to (77.120,135.003) but not as close as when we considered the mean parameter. The method of taking the quantiles from the chains of the distribution is known as the percentile method in the bootstrapping literature (see Efron & Tibshirani (1993) for more details). This technique is known to be biased for parameters with skewed distributions and with small samples, and other methods for example the BCA method can be used instead, but they will not be discussed here.

Nonparametric bootstrapping is another stochastic estimation technique that we can apply to our problem. Here we do not assume a distribution for the data but instead generate a large number of data sets by sampling (with replacement) from the original sample. In our example we will again generate samples of size 100 using a macro. This macro is stored in the file **hboot.txt** as shown:


C:\Program Files (x86)\MLwiN v2.31\samples\hboot.txt
seed 1
note set random number seed to 1
erase c5 c6
note erase all data in columns c5 and c6
loop b1 1 10000
note repeat the following commands 10,000 times
boot 100 c1 c2
note pick 100 values with replacement from column c1 and put in column c2
aver c2 b2 b3 b4
note calculate the mean (b3) and standard deviation(b4) of the 100 draws
calc b4=b4*b4
note calculate the variance(b4) of the 100 draws.
join c5 b3 c5
join c6 b4 c6
note store the mean in c5 and the variance in c6
endl
name c5 'npmean'
name c6 'npvar'
note name the mean and variance columns
Go to end
Execute
Find
Execute selection
Replace

This macro is rather similar to the macro for parametric bootstrapping except for the method of generating samples. In the earlier macro we used the NRAN and CALC commands to construct each bootstrap data set from the correct Normal distribution. Here we use the BOOT command:

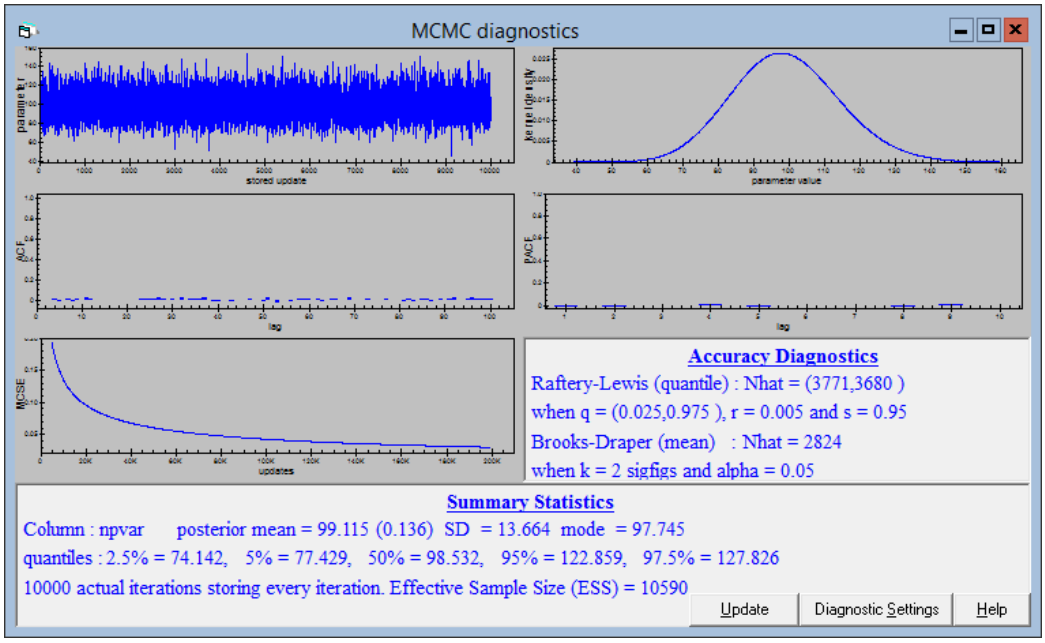
```
► BOOT 100 c1 c2
```

This command constructs a sample of size 100 in column **c2** by sampling with replacement from the values in column **c1**.

Open the **hboot** macro as before, click on **Execute** and the macro will run putting the 10,000 means into column **c5** named **npmean** and the 10,000 variances into column **c6** named **npvar**.

Again we can look at the summaries of both of these parameters using the **Column diagnostics** window. The summary for the mean is not shown here, but it gives a central interval of (173.37, 177.32) for the population mean. This is approximately equal to the Normal theory interval of (173.39, 177.31) expected from the central limit theorem.

Let's look at the variance parameter. Use the **Column Diagnostics** window as before to select **npvar** for analysis. We obtain the following display:


MCMC diagnostics
parameter
stored update
kernel density
parameter value
ACF
lag
PACF
lag
MCSE
updates
Accuracy Diagnostics
Raftery-Lewis (quantile) : Nhat = (3771,3680)
when q = (0.025,0.975), r = 0.005 and s = 0.95
Brooks-Draper (mean) : Nhat = 2824
when k = 2 sigfigs and alpha = 0.05
Summary Statistics
Column : npvar posterior mean = 99.115 (0.136) SD = 13.664 mode = 97.745
quantiles : 2.5% = 74.142, 5% = 77.429, 50% = 98.532, 95% = 122.859, 97.5% = 127.826
10000 actual iterations storing every iteration. Effective Sample Size (ESS) = 10590
Update
Diagnostic Settings
Help

Here we again see that the kernel density plot shows a slight skew to the right. The 95% central confidence interval from the percentile method is (74.142,127.826). This is only slightly different from the interval from the parametric method, which shows that the Normality assumption is acceptable in this case.

Having seen this simple illustration of simulation while knowing how easy it is to compute theoretical intervals for the mean and variance of a Normal distribution, you may be asking yourself what benefit there is in using simulation methods such as bootstrapping. As we have seen in earlier chapters, estimation of multilevel model parameters is far more complex than the computations used in this example. In most multilevel models iterative routines are required to get estimates, and no simple formulas exist for the distributions of the parameters. In situations like this, simulation techniques come into their own. Using MCMC and bootstrapping methods, it is often easy to generate simulated values from the distributions of the parameters of interest and hence calculate point and interval estimates.

16.2 Generating random numbers in MLwiN

The above height data set was actually generated by simulating from a known Normal distribution and rounding the heights to the nearest cm. MLwiN allows you to generate random numbers from several common distributions. To demonstrate this we will consider another example. Here we will still consider a data set of people's heights but this time we will consider a mixed population of men and women. We will assume that the men are Normally distributed with a mean of 175cm and a standard deviation of 10cm while the women have a mean of 160cm and a standard deviation of 8cm. We will

also assume that men make up 40% of the population while women make up the other 60%. We will now show how to use the Generate Random Numbers window in MLwiN to create a sample of 100 people from this population.

We will first generate 100 standard Normal random numbers from which we will construct our sample.

- Open the **Generate Random Numbers** window from the **Basic Statistics** menu
- In the **Type of Number** frame, select **Normal Random Number**
- Set the **Output column** to **c7**
- Set the **Number of repeats** to **100**
- Click **Generate**

We now have the 100 random draws in column **c7**. (Incidentally this is equivalent to the NRAN command used in the parametric bootstrap macro). We now want to generate another 100 random numbers, this time from a Binomial distribution, to represent whether the person is male or female. To do this, use the **Generate Random Numbers** window again.

- In the **Type of Number** frame, select **Binomial Random Number**
- Set the **Output column** to **c8**
- Set the **Number of repeats** to **100**
- Set the **Probability** to be **0.6**
- Set the **Number of Trials** to be **1**
- Click **Generate**

We can now name columns **c7 'Normal'** and **c8 'Female'** using the **Names** window. Next we open the **Calculate** window from the **Data Manipulation** menu, and create another variable **'Male'** in **c9** that is equal to 1 − 'Female' as follows:

- In the large calculation space on the right side, enter the following:

```
► c9 = 1 - 'Female'
```

- Click on **Calculate**
- Name **c9** 'Male' using the **Names** window

We can now at last construct the actual height variable (c10) in a similar way using the following formula:

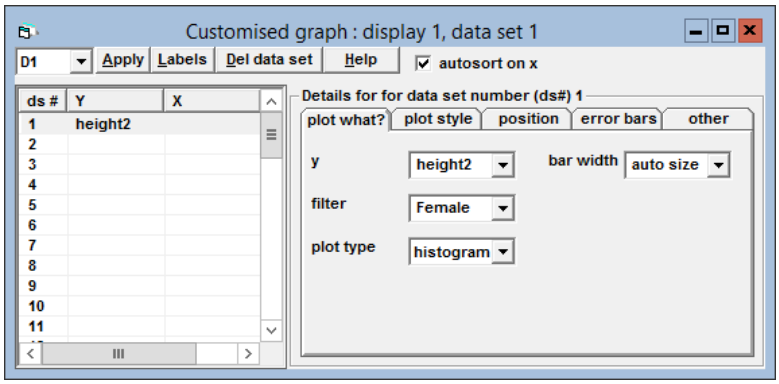
- Type the following:

```
► c10 = (175 + 'Normal' * 10) * 'Male' + (160 +
  'Normal' * 8) * 'Female'
```

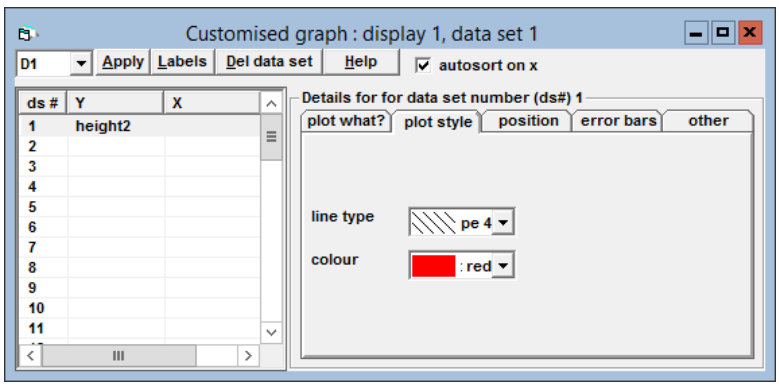
- Click on **Calculate**

We will give column c10 the name **height2** using the **Names** window. We can now construct a histogram to look at our data set. In this plot we will plot the males and females in different colours to show the composition of the mixture model.

Open the **Customised graph** window and choose display **D1** to construct the histogram. (If you already have a graph set up on display **D1** then you can either delete any existing data sets or use a different display number.) We will first plot the female heights. To do this, select data set **ds#1** on the **plot what?** tab and make selections as follows:

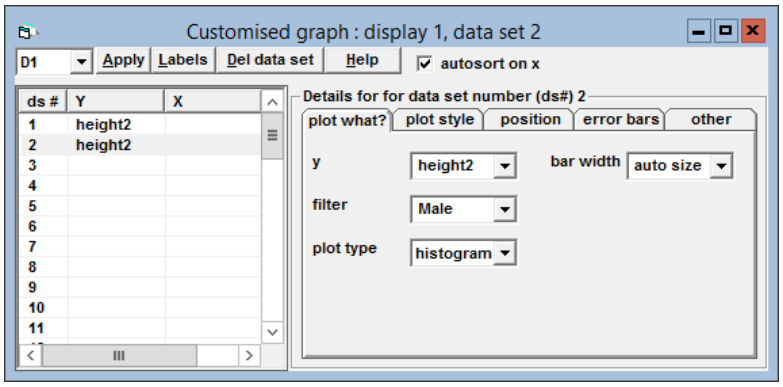


The filter option tells the software to only plot points when female is equal to 1. We also want to set different styles for the female and male heights so we set up the **plot style** tab as follows:

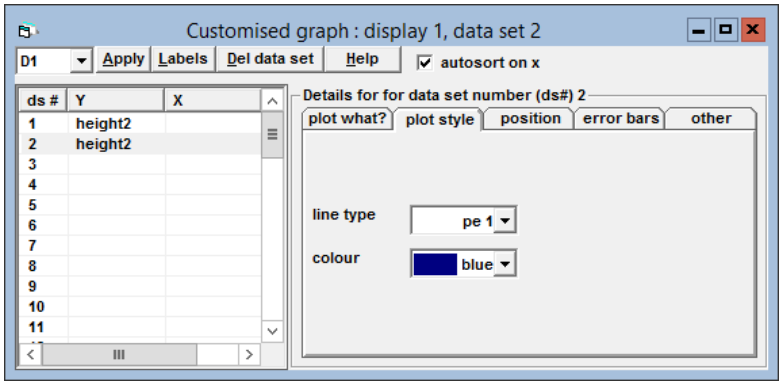


We now need to add the male heights to the graph. To do this we assign the male heights to data set **ds#2** to be plotted on the same graph. The **plot**

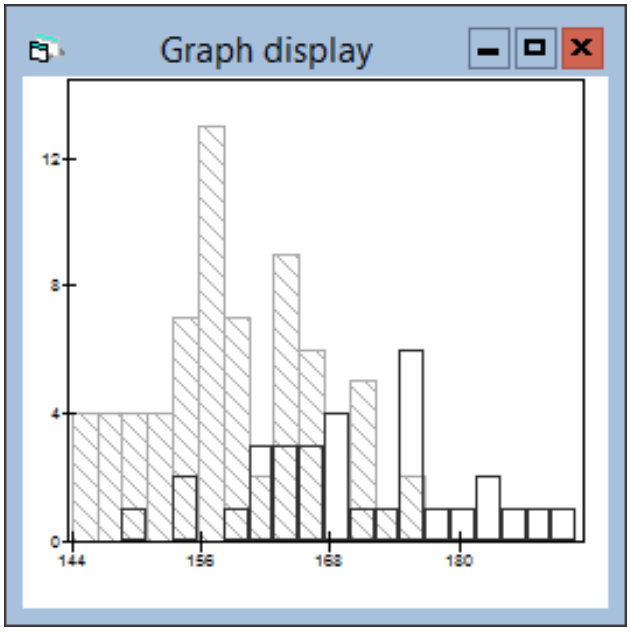
what? tab should be set as follows:



Again we need to set the **plot style** tab for this new graph. This should be done as follows:



Having completed the set up of the two data sets, we can click the **Apply** button to view the histogram. Since random numbers were used to generate the data displayed in the histogram, your graph may differ from the one shown here:



This graph clearly shows the two distributions from which our heights have been generated. In this example we have the sex of each individual whose height was measured and so we could quite reasonably fit separate Normal models for males and females and construct interval estimates for each separately. It may, however, be the case that we do not know the sex of the individuals, and here we could use a nonparametric bootstrapping method to model the data set.

Chapter learning outcomes

- ⋆ That simulation-based methods can be used as an alternative to the iterative methods used so far to fit multilevel models
- ⋆ How to use bootstrapping to fit simple models
- ⋆ How to generate random numbers in MLwiN
- ⋆ How to execute macros in MLwiN
- ⋆ How to create histograms in MLwiN

Chapter 17

Bootstrap Estimation

17.1 Introduction

We have already introduced the idea of bootstrapping[1] for a simple one-level problem. In multilevel modelling the bootstrap can be used as an alternative to MCMC estimation for two main purposes: (1) improving the accuracy of inferences about parameter values and (2) correcting bias in parameter estimates.

With continuous response models we can construct confidence intervals for functions of the fixed parameters by assuming Normality of the random errors, but this approach may not be appropriate for the random parameters unless the number of units at the level to which the parameter refers is large. Bootstrapping provides an improved procedure for constructing confidence intervals for random parameters.

Bootstrap estimation is useful in fitting models with discrete responses where the standard quasilikelihood-based estimation procedure produces estimates — especially of the random parameters — that are downwardly biased when the corresponding number of units is small. (See Goldstein & Rasbash (1996) for a discussion of this problem.) The severity of this bias can be trivial in some data sets and severe in others. A complicating factor in fitting these models is that the bias is a function of the underlying 'true' value, so that the bias correction needs to be iterative. In the next section we illustrate how this works.

In Chapter 16 we saw how bootstrapping was used to construct several simulated data sets from the original data set and/or model parameters. Then estimates of the parameters of interest were found for each of these new data sets, creating a chain of values that allowed parameter estimates' distribu-

[1] For a more complete introduction to bootstrapping, see Efron & Tibshirani (1993).

tional summaries to be obtained. In multilevel modelling, the implementation of bootstrapping is similar. The bootstrapping methods are used to construct the bootstrap data sets, and then either the IGLS or RIGLS estimation method is used to find parameter estimates for each data set. The parametric bootstrap works exactly as in Chapter 16 in that the data sets are generated (by simulation) based on the parameter estimates obtained for the original data set. Due to the multilevel structure of the data modelled with MLwiN, however, we cannot use the simple nonparametric approach introduced in Chapter 16, but instead we will introduce a new method based on sampling from the estimated residuals.

17.2 Understanding the iterated bootstrap

Suppose we simulate a data set for a simple variance components model where the true value for the level 2 variance, σ_u^2, is 1.0. Suppose also that the standard MLwiN estimation procedure has a downward bias of 20% for the level 2 variance parameter. If we fit this model for several simulated data sets using the standard procedure we will obtain an average estimate of 0.8 for this parameter.

Imagine now that we have just one simulated data set with a level 2 variance estimate that happens to be 0.8, together with fixed parameter estimates to which we can apply the same procedure. We can now simulate (parametrically bootstrap) a large number of new response vectors from the model with level 2 variances of 0.8 and calculate the average of the variance estimates across these new replicates. We would expect a value of 0.64 since the level 2 variance is estimated with a downward bias of 20% ($0.8 \times 0.8 = 0.64$). Now if we add the downward bias of 0.16 to our starting value of 0.8, we obtain a *bias corrected* estimate of 0.96. We can now run another set of simulations, this time taking the bias corrected estimates (0.96 for the variance) as our starting simulation values. After fitting the model to each of these new replicates we expect an average of 0.768 for the variance parameter. This results in a bias estimate of 0.192. We then add this estimated bias to 0.8 to give a bias corrected estimate of 0.992. We can now go on to simulate yet another set of replicates using the latest bias corrected estimate and repeat until the successive corrected estimates converge (see the table below). We shall see how we can judge convergence in the example that follows. Note that in models for which the bias is independent of the underlying true value (additive bias), only a single set of bootstrap replicates is needed for bias correction.

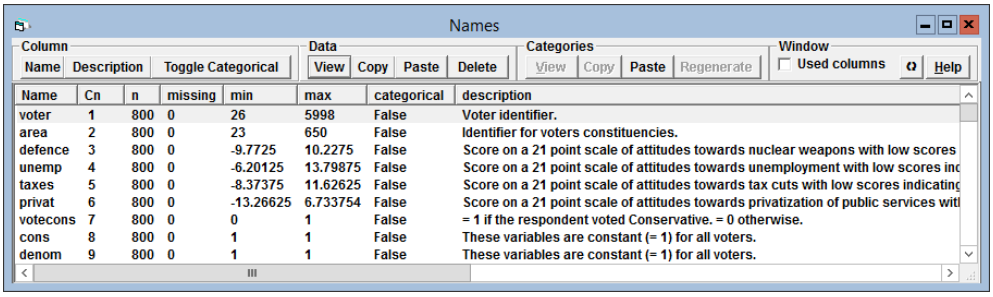
Replicate Set	Starting Value	Simulated Estimate (Standard procedure)	Estimate (Bias corrected)
1	0.8	0.8*0.8 = 0.64	0.8+(0.8-0.64) = 0.96
2	0.96	0.96*0.8 = 0.768	0.8+(0.96-0.768) = 0.992
3	0.992	0.992*0.8 = 0.7936	0.8+(0.992-0.7936) = 0.9984
4	0.9984	0.9984*0.8 = 0.7987	0.8+(0.9984-0.7987) = 0.9997
5	0.9997	0.9997*0.8 = 0.7997	0.8+(0.9997-0.7997) = 1.0000

Up to the time of this release of MLwiN, the user community still has relatively little experience in using bootstrap methods with multilevel models. We suggest therefore that this procedure should be used with care. Bootstrap estimation is based on simulation and therefore convergence is stochastic. This raises the question of what is a large enough number of replicates in each bootstrap *set*. On the examples tried, sets of between 300 and 1000 replicates and a series of about five sets is usually sufficient to achieve convergence. The total process thus involves a substantial amount of computation. For this reason, bootstrapping, like MCMC estimation should not be used for model exploration, but rather to obtain unbiased estimates and more accurate interval estimates at the final stages of analysis.

At convergence, the current replicate set can be used to generate confidence intervals or any other desired descriptive statistic for model parameters (see below).

17.3 An example of bootstrapping using MLwiN

The data for the example come from the longitudinal component of the British Election Study (Heath et al., 1996). The data set contains records from a subsample of 800 voters grouped within 110 voting constituencies who were asked how they voted in the 1983 British general election. For our purposes, the response variable has been categorized as having voted Conservative or not. Open the worksheet **bes83.ws**. The **Names** window shows the following variables:

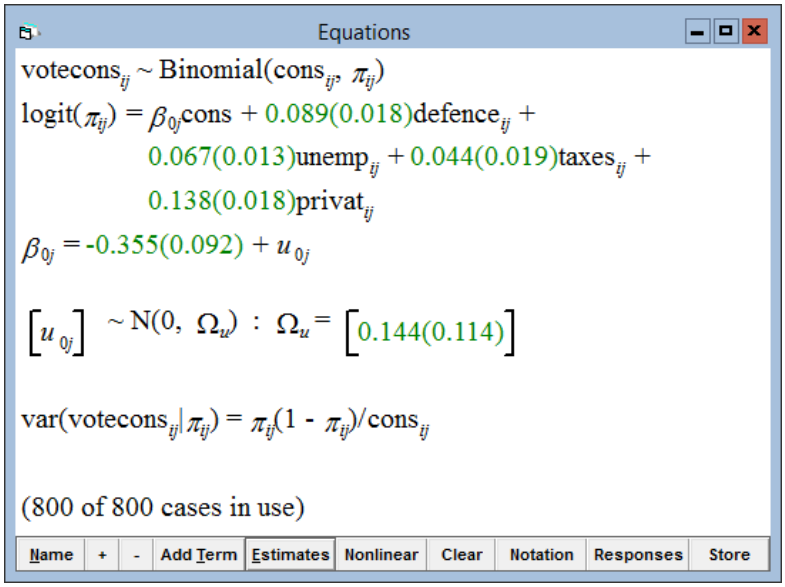


Name	Cn	n	missing	min	max	categorical	description
voter	1	800	0	26	5998	False	Voter identifier.
area	2	800	0	23	650	False	Identifier for voters constituencies.
defence	3	800	0	-9.7725	10.2275	False	Score on a 21 point scale of attitudes towards nuclear weapons with low scores
unemp	4	800	0	-6.20125	13.79875	False	Score on a 21 point scale of attitudes towards unemployment with low scores inc
taxes	5	800	0	-8.37375	11.62625	False	Score on a 21 point scale of attitudes towards tax cuts with low scores indicating
privat	6	800	0	-13.26625	6.733754	False	Score on a 21 point scale of attitudes towards privatization of public services witl
votecons	7	800	0	0	1	False	= 1 if the respondent voted Conservative. = 0 otherwise.
cons	8	800	0	1	1	False	These variables are constant (= 1) for all voters.
denom	9	800	0	1	1	False	These variables are constant (= 1) for all voters.

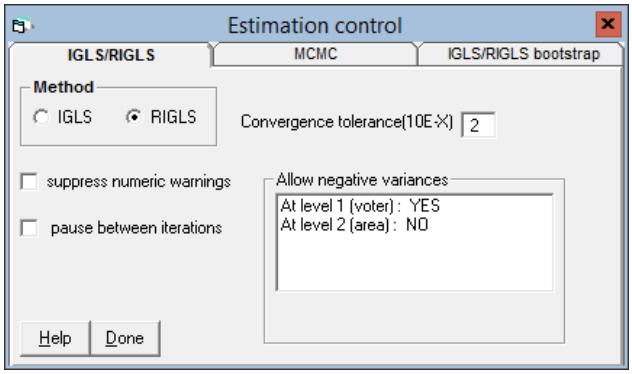
These variables are defined as follows:

Variable	*Description*
voter	Voter identifier
area	Identifier for voters' constituencies
defence	Score on a 21 point scale of attitudes towards nuclear weapons with low scores indicating disapproval of Britain possessing them. This variable is centred about its mean.
unemp	Score on a 21 point scale of attitudes towards unemployment with low scores indicating strong opposition and higher scores indicating a preference for greater unemployment if it results in lower inflation. This variable is centred about its mean.
taxes	Score on a 21 point scale of attitudes towards tax cuts with low scores indicating a preference for higher taxes to pay for more government spending. This variable is centred about its mean.
privat	Score on a 21 point scale of attitudes towards privatization of public services with low scores indicating opposition. This variable is centred about its mean.
votecons	= 1 if the respondent voted Conservative = 0 otherwise
cons	These variables are constant (= 1) for all voters
bcons	These variables are constant (= 1) for all voters
denom	These variables are constant (= 1) for all voters

Begin by setting up a two-level variance components model, with **voter** as the level 1 identifier, **area** as the level 2 identifier, **votecons** as the response variable and **cons**, **defence**, **unemp**, **taxes** and **privat** as explanatory variables. Refer to Chapter 9 if you need detailed assistance in doing this. If you fit this model using first order MQL, RIGLS estimation, you will obtain the following results:

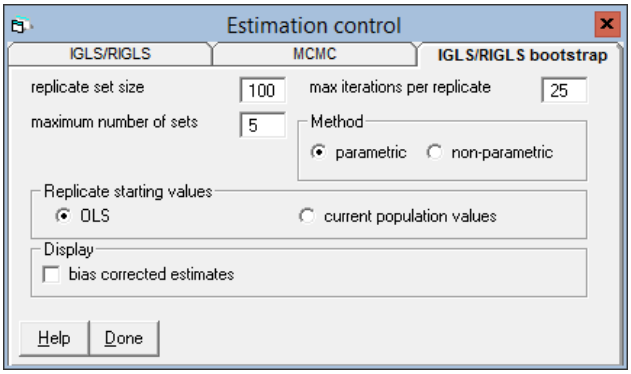

Equations
votecons$_{ij}$ ~ Binomial(cons$_{ij}$, π_{ij})
logit(π_{ij}) = β_{0j}cons + 0.089(0.018)defence$_{ij}$ +
0.067(0.013)unemp$_{ij}$ + 0.044(0.019)taxes$_{ij}$ +
0.138(0.018)privat$_{ij}$
β_{0j} = -0.355(0.092) + u_{0j}
$\begin{bmatrix} u_{0j} \end{bmatrix}$ ~ N(0, Ω_u) : Ω_u = $\begin{bmatrix} 0.144(0.114) \end{bmatrix}$
var(votecons$_{ij}$|π_{ij}) = π_{ij}(1 - π_{ij})/cons$_{ij}$
(800 of 800 cases in use)
Name + - Add Term Estimates Nonlinear Clear Notation Responses Store

You may wish to experiment with a range of bootstrapping options using this as the base model, so save this model in a worksheet so you can return to it at a later stage. To set up a bootstrap run, first click on the **Estimation control** button on the **main toolbar** and the following window will appear:



In the **Allow negative variances** box, click on **At level 2 (area)** to change **NO** to **YES**. If it does not already say **YES** next to **At level 1 (voter)** click on this as well to change it. This will allow negative variances at both levels. This means that any negative variances that occur in individual bootstrap replicates will be retained, rather than set to zero, so that a consistent bias correction is estimated.

We can now click on the **IGLS / RIGLS bootstrap** tab to display the following:



For the first analysis, select the **parametric** method of bootstrapping. (We will discuss the nonparametric bootstrap in a later section.) Here we can also set the number of replicates per set and the maximum number of sets. We can also set the maximum number of iterations per replicate.

If a replicate has not converged after the specified number of iterations (these are standard MLwiN iterations) the replicate is discarded. MLwiN does not judge a set of replicates to be completed until the full quota of converged replicates has been run. While the bootstrap is running, a progress count is maintained at the bottom of the screen, and the number of discarded replicates is also reported. If this count grows large you may wish to restart the bootstrap with a higher setting for maximum number of iterations per replicate. We will initially use the displayed default settings, so now click on the **Done** button.

We want to watch the progress of the bootstrap as estimation proceeds, and we can do so using the **Trajectories** window. The parameter whose estimate exhibits the most bias is the level 2 (between-area) variance. We will set the **Trajectories** window to show the graph for this parameter only. Note that opening the Trajectories window will slow down the iterations.

- Open the window by selecting **Trajectories** from the **Model** menu.
- Click the **Select** button and choose **area:cons/cons** from the **Select plots** list that appears
- Select **1 graph per row** from the drop-down list at the bottom right of the window
- Click **Done**

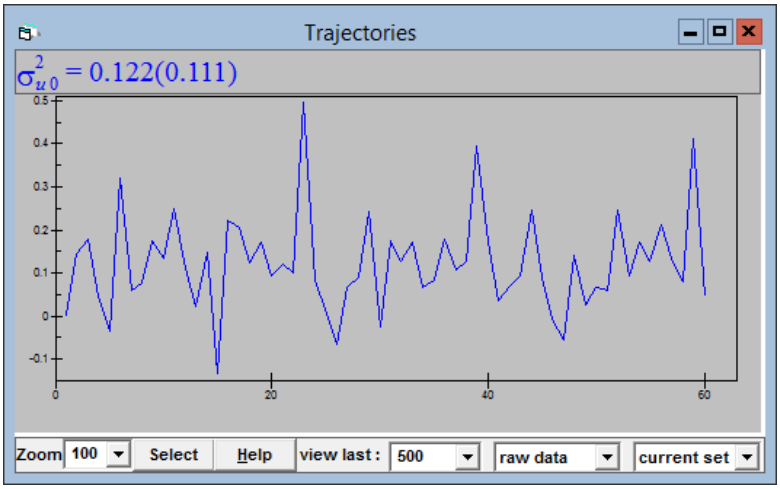
All the bootstrap runs shown in this chapter were run with a seed value of 100 for the random number generator. If you wish to produce exactly the same results, you can set the random number seed by opening the **Command interface** window and typing the command:

```
► seed 100
```

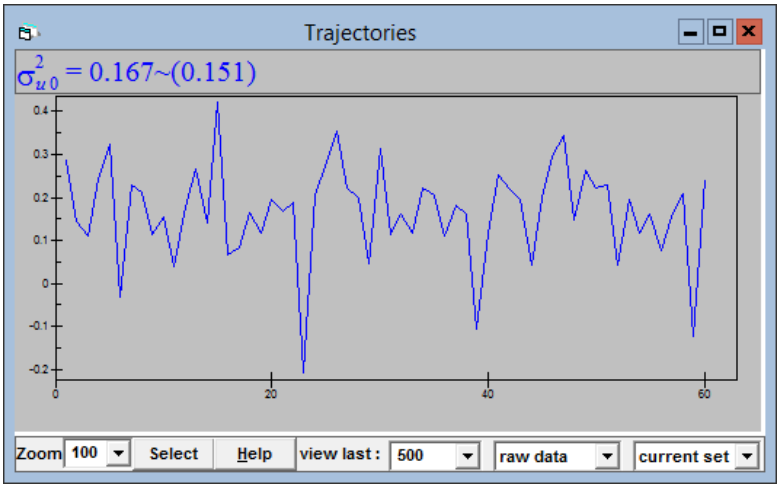
You may prefer to set a different seed value or to let MLwiN choose its own seed value.

> *Note that it is important that you do not change any other settings for the model after running to convergence using quasilikelihood, e.g., switching from PQL to MQL.*

We can now set the bootstrap running by clicking the **Start** button. The **Trajectories** window will display the bootstrap chain for the current replicate set. After approximately 60 replicates the bootstrap chain for the first replicate set will look somewhat like this:



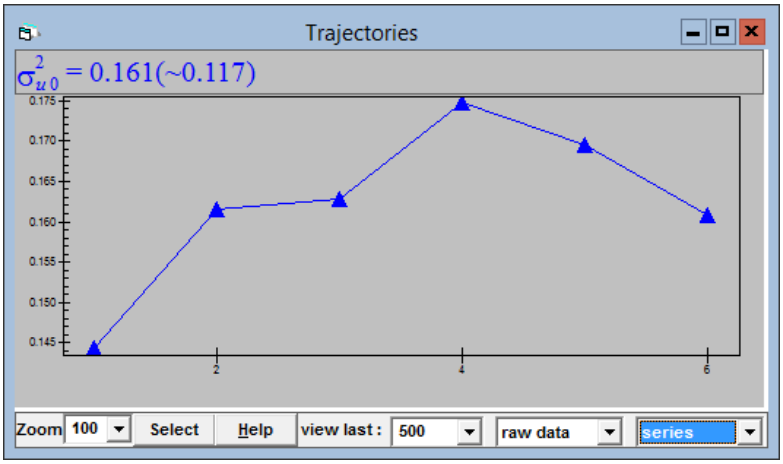
Note that since we are allowing negative variances, the sequence of estimates crosses the x-axis. By default the values shown are not corrected for bias. We can at any time switch between bias corrected and uncorrected estimates by opening the **Estimation control** window and checking the **bias corrected estimates** box. When this box is checked, another checkbox labelled **scaled SE's** appears. If you check both these boxes you will observe that the **Trajectories** window changes as follows:



The current bootstrap estimate increases, in this case, from 0.123 to 0.165. Note that we started with the RIGLS estimate of 0.144 and hence the bias corrected estimate is simply calculated as $0.144 + (0.144 - 0.123) = 0.165$. The **scaled SE's** option only changes the reported standard error shown in brackets above the graph. This scaling process ensures that standard errors and quantile estimates for bias corrected estimates are properly scaled. The scaling is an approximation and hence scaled standard errors and quantiles are preceded by a tilde ($\sim$). See the **Help** system for more details.

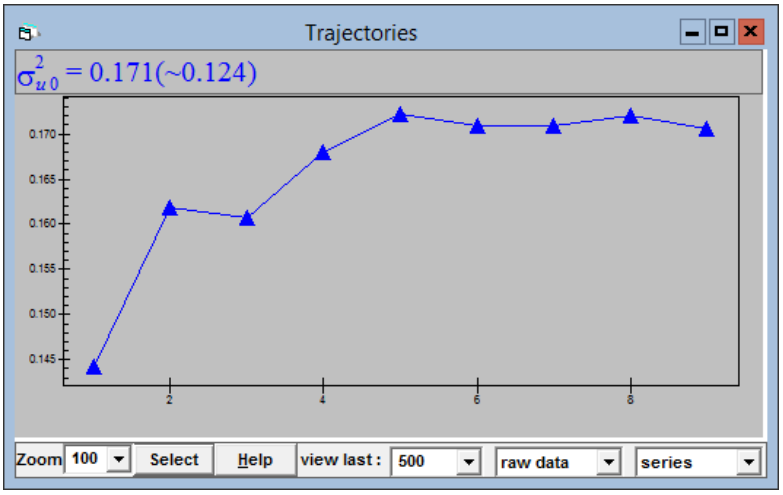
It is useful to view bootstrap replicate sets as a sequence of running means because this gives some clues about convergence. You can do this by selecting **running mean** on the middle drop-down list box at the bottom of the **Trajectories** window. The previous figure shows **raw data** (the default) currently selected. A converged running mean chain should be reasonably stable.

At any point you can change from viewing the current replicate set chain to the series of replicate set summaries by selecting **series** on the right hand drop-down list on the **Trajectories** window's tool bar. (The **current set** option is the default.) When viewing the series of set summaries, it is more informative to select **raw data** rather than **running mean**. After five sets of bootstrap replicates, the series graph looks like this:



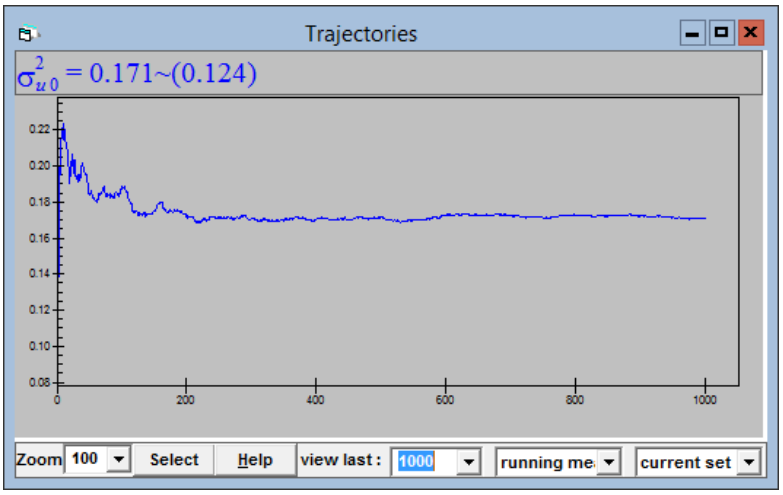
This graph is useful for judging bootstrap convergence. If we are viewing bias corrected estimates we would expect to see this graph levelling out if the bootstrap series has converged. This is not the case here. This means we probably need to increase the replicate set size to reduce simulation noise.

The following figure shows the same type of series graph for a bootstrap run with this data set and model but with the replicate set size increased from 100 to 1000 and the number of replicate sets increased from five to eight. Note that making these increases will result in a greatly extended running time for the full bootstrap process — up to several hours in this example.



As we might have expected, this graph is less erratic and we can therefore have more confidence that the bootstrap has converged. In fact it looks reasonably stable after set 4. From an original estimate of 0.144, the bootstrap process eventually produces a bias corrected estimate of 0.171.

The complete running mean sequence for the last replicate set of this run appears as follows:

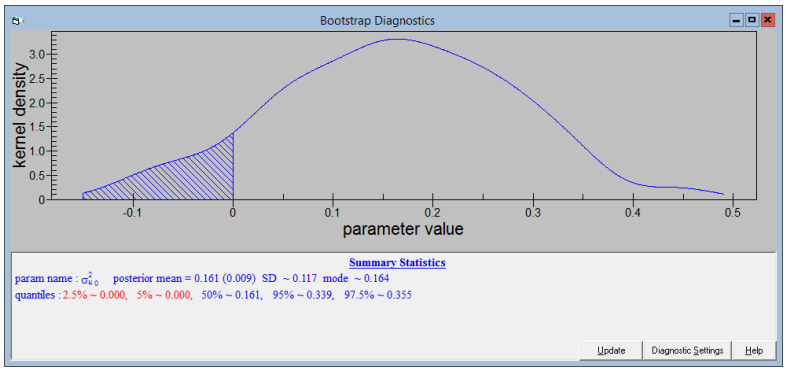


To display this graph, the **running mean** and **current set** options were selected from the middle and right pull down lists, respectively, and the number in the **view last:** box was increased to 1000. We see that this replicate set's result stabilised after between 200 and 400 replicates. This means that in this example we should have been able to use a replicate set size of 400 and a series of five sets. It is generally sensible, however, to be cautious in selecting replicate set sizes and series lengths.

17.4 Diagnostics and confidence intervals

At any stage in the bootstrap process, when viewing a replicate set chain we can obtain a kernel density plot and quantile estimates that are calculated from the chain. This is achieved by clicking on the graph in the **Trajectories** window for the parameter we are interested in.

The figure below shows the diagnostics obtained for a (bias corrected) replicate set involving 100 replicates:



Notice first that there is some irregularity in the kernel density plot due to having too few replicates per set. In this case 500 may be a more suitable number. The second thing to notice is that the area below zero on the x-axis is shaded. This occurs because we are viewing a kernel density for estimates of a variance parameter, and these are usually positive. Although we allow negative estimates for variances during bootstrap estimation to ensure consistent bias correction, when we calculate quantiles and summary statistics, we set to zero any results that are negative.

17.5 Nonparametric bootstrapping

When we considered the single-level example in Chapter 16 with a sample of 100 heights, it was easy to perform a nonparametric bootstrap. We simply drew samples of size 100 with replacement from the 100 heights. When we consider a multilevel model where the responses come from different higher-level units, an analogous approach is problematic.

Consider the tutorial example covered in the first section of the manual, where we had 4059 students in 65 schools. If we were simply to sample pupils with replacement then we would generate data sets that do not have the same structure as our original data set. For example, although a particular school may actually have 10 pupils, in the first simulated data set it could have 15 pupils generated for it, or even worse, no pupils. On the other hand, sampling with replacement within each school is problematic because some

schools have very few pupils.

The approach used in MLwiN involves resampling from the estimated residuals generated from the model (as opposed to the response variable values). It may be referred to more precisely as a semi-parametric bootstrap since the fixed parameter estimates are used. The procedure incorporates sampling from the unshrunken residuals to produce the correct variance estimates. The procedure is included here for completeness and is also described in the **Help** system.

The resampling procedure

Consider the two-level model

$$\begin{aligned} y_{ij} &= (X\beta)_{ij} + (ZU)_j + e_{ij} \\ U^T &= \{U_0, U_1, \ldots\} \end{aligned}$$

Having fitted the model, we estimate the residuals at each level as

$$\hat{U} = \{\hat{u}_0, \hat{u}_1, \ldots\}, \hat{e}$$

If we were to sample these residuals directly, we would underestimate the variance parameters because of the shrinkage. It is the case that the correlation structure among the estimates within and between levels reproduces the correct total variance when estimated residuals are used. However the random sampling with replacement upon which bootstrap sampling of the residuals is based will not preserve this structure and so will not generally produce unbiased estimates of either the individual random parameters or of the total variance and covariance of responses.

One possibility, shown already in this chapter, is to use a fully parametric bootstrap. This, however, has the disadvantage of relying upon the Normality assumption for the residuals. Instead we can resample estimated residuals to produce unbiased distribution function estimators, as follows.

For convenience we shall illustrate the procedure using the level 2 residuals, but analogous operations can be carried out at all levels. Write the empirical covariance matrix of the estimated residuals at level 2 in model (1) as

$$S = \frac{\hat{U}^T\hat{U}}{M}$$

and the corresponding model estimated covariance matrix of the random coefficients at level 2 as R. The empirical covariance matrix is estimated using the number of level 2 units, M, as divisor rather than $M - 1$. We assume that the estimated residuals have been centred, although centring will only affect the overall intercept value.

We now seek a transformation of the residuals of the form

$$\hat{U}^* = \hat{U}A$$

where A is an upper triangular matrix of order equal to the number of random coefficients at level 2, and such that

$$\hat{U}^{*T}\hat{U}^* = A\hat{U}^T\hat{U}A = A^TSA = R$$

The new set of transformed residuals $\hat{U}^*$ now have covariance matrix equal to the one estimated from the model, and we sample sets of residuals with replacement from $\hat{U}^*$. This is done at every level of the model, with sampling being independent across levels.

To form A we note that we can write the Cholesky decomposition of S in terms of a lower triangular matrix as $S = L_S{L_S}^T$ and the Cholesky decomposition of R as $R = L_R{L_R}^T$. We then have

$$L_R{L_S}^{-1}\hat{U}^T\hat{U}(L_R{L_S}^{-1})^T = L_R{L_S}^{-1}S(L_S^{-1})^T(L_R)^T = L_R(L_R)^T = R$$

Thus, the required matrix is

$$A = (L_R{L_S}^{-1})^T$$

and we can hence find the $\hat{U}^* = \hat{U}A$ and then use them to bootstrap a new set of level 2 residuals. MLwiN automatically carries out these calculations when using the nonparametric procedure.

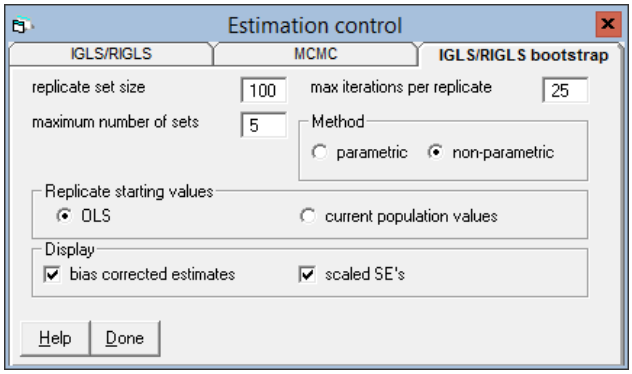
Example using the British Election Study data set

Although the nonparametric and parametric bootstrap procedures differ in their methods for creating the bootstrap data sets, they both produce chains of parameter estimate values. We will now repeat our analysis of the **bes83.ws** data set using the nonparametric bootstrap. The seed value of 100 is again

used for the random number generator to produce the results shown in this section.

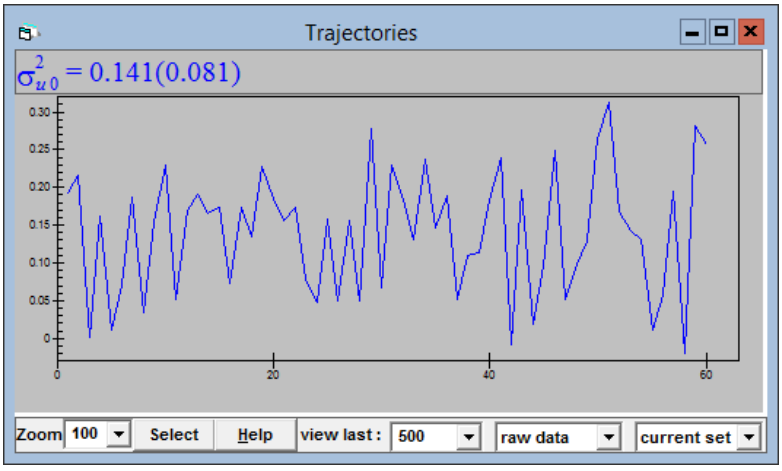
To start the example, retrieve the worksheet, set up the model as in Section 17.3 and fit it using first order MQL RIGLS estimation. Note that if you have just worked through the parametric bootstrap example, the model is already set up and you simply need to change estimation method to RIGLS and refit the model.

We now want to select the bootstrap method. Click on the **Estimation control** button on the main toolbar and the **IGLS / RIGLS** options will appear in the **Estimation control** window. Ensure that you set both levels to **YES** in the **Allow negative variances** box before you click on the **IGLS / RIGLS bootstrap** tab. Now select the **nonparametric bootstrap** option from the **Method** box. We will leave the other parameters at their default values so that the window appears as below. Having set the parameters, click on **Done** to continue.

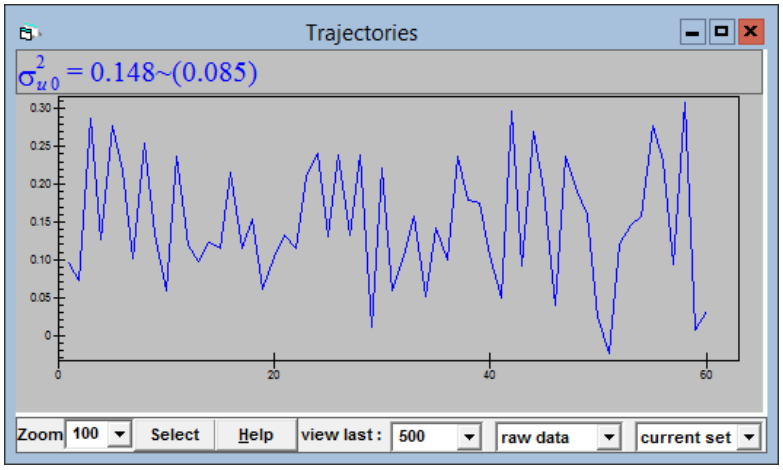


We can now repeat the analysis performed in Section 17.3 with the nonparametric bootstrap.

Click on the **Start** button on the main toolbar to set the nonparametric bootstrap running. After 60 or so replicates, the bootstrap chain for the first replicate set of the level 2 variance parameter should look like this:

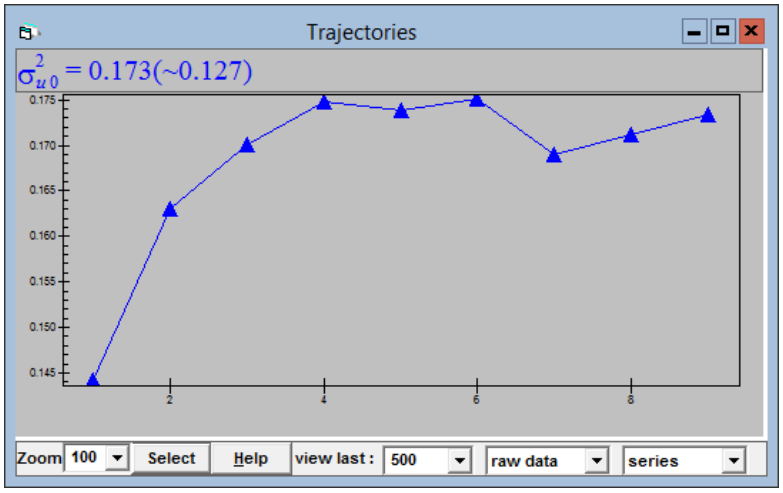


We can again switch from uncorrected to bias corrected estimates by checking the **bias corrected estimates** box in the **Estimation control** window. Doing so and selecting **scaled SEs** will transform the **Trajectories** window as follows:

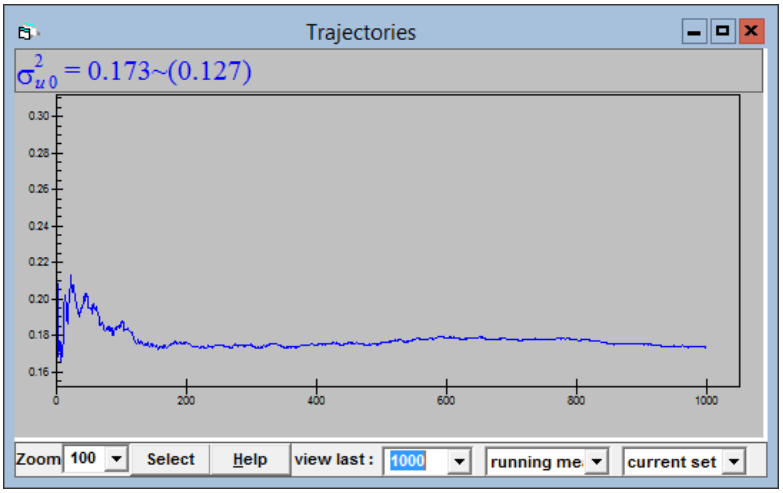


In Section 17.3 we saw that running five sets of 100 iterations was not enough for the parametric bootstrap. A similar result will be seen for the nonparametric procedure. We will skip the illustration of this finding and instead immediately increase the replicate set size to 1000 and the number of replicate sets to eight. Note again that to run the bootstrap with this model and data set takes several hours on most computers.

The graph below shows a plot of the (bias corrected) series means for the level 2 variance parameter using nonparametric bootstrapping. This graph compares favourably with the graph for the parametric bootstrap, which had final estimate 0.171(~0.124).

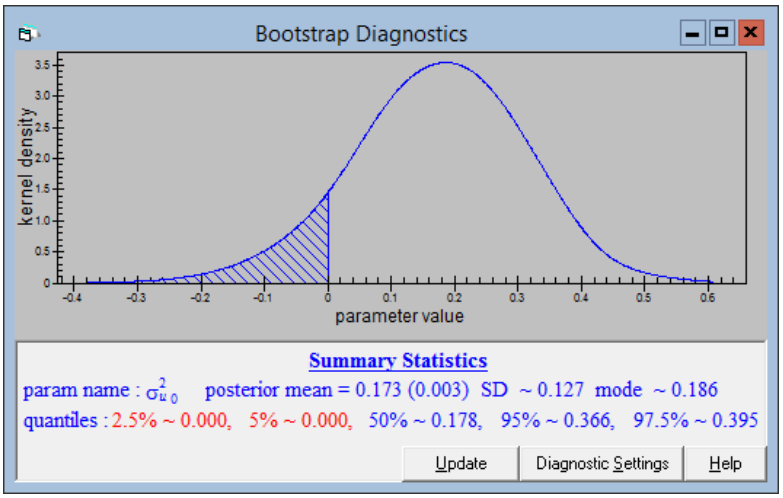


Here we see that the graph looks fairly stable after step 3, and from an original estimate of 0.144 the bootstrap eventually produces a bias corrected estimate of 0.173. The graph of the running mean sequence for the last replicate set on this bootstrap is as follows:



We see that this replicate set has stabilised by the time 200 replicates have been reached. This means that for the nonparametric bootstrap on this example we could probably have used a replicate set size of 250 and a series of 5 sets. It is generally sensible, however to select replicate set size and series lengths that are conservative.

To use the last chain to get interval estimates for this parameter, we first need to select raw data instead of running mean from the middle drop-down option box at the bottom of the **Trajectories** window. This will bring up the actual chain of 1000 bootstrap replicates in the final replicate set. Now clicking on the **Trajectories** window display will bring up the **Bootstrap Diagnostics** window as shown below.



Here we see that running 1000 replicates produces a smoother curve than seen in the earlier plot for the parametric bootstrap based on 100 replicates. We again see here that both the 2.5% and 5.0% quantiles are estimated as zero as they correspond to negative values in the kernel density plot. This means that for our model, the level 2 variance is not significantly different from zero.

Chapter learning outcomes

- ⋆ How to use the bootstrap options in MLwiN
- ⋆ How the iterative bootstrap gives unbiased estimates
- ⋆ The difference between parametric and nonparametric bootstrapping.

Chapter 18

Modelling Cross-classified Data

18.1 An introduction to cross-classification

An important motivation for multilevel modelling is the fact that most social processes we wish to model take place in the context of a hierarchical social structure. The assumption that social structures are purely hierarchical, however, is often an over-simplification. People may belong to more than one grouping at a given level of a hierarchy and each grouping can be a source of random variation. For example, in an educational context both the neighbourhood a child comes from and the school a child goes to may have important effects. A single school may contain children from many neighbourhoods and different children from any one neighbourhood may attend several different schools. Therefore school is not nested within neighbourhood and neighbourhood is not nested within school: instead, we have a cross-classified structure. The consequences of ignoring an important cross-classification are similar to those of ignoring an important hierarchical classification.

A simple model in this context can be written as:

$$y_{i(jk)} = \alpha + u_j + u_k + e_{i(jk)} \tag{18.1}$$

where the achievement score $y_{i(jk)}$ of the ith child from the (jk)th school / neighbourhood combination is modelled by the overall mean α, together with a random departure u_j due to school j, a random departure u_k due to neighbourhood k, and an individual-level random departure $e_{i(jk)}$.

The model can be elaborated by adding individual-level explanatory variables, whose coefficients may also be allowed to vary across schools or neighbourhoods. Also, school or neighbourhood level variables can be added to explain variation across schools or neighbourhoods.

Another type of cross-classification occurs when each pupil takes a single exam paper that is assessed by a set of raters. If a different set of raters operates in each school, we have a pupil / rater cross-classification at level 1 nested within schools at level 2. A simple model for this situation can be written as:

$$y_{(ij)k} = \alpha + u_k + e_{ik} + e_{jk} \tag{18.2}$$

where the rater and pupil effects are modelled by the level 1 random variables e_{ik} and e_{jk}. The cross-classification need not be balanced, and some pupils' papers may not be assessed by all the raters.

Yet another example involves repeated measures. Suppose a sample of different veterinarians measured the weights of a sample of animals, each animal being measured once. If independent repeat measurements were made by the vets on each animal, this would become a level 2 cross-classification with replications within cells. In fact, we could view the first case as a level 2 classification where there just happened to be only one observation per cell. Many cross-classifications will allow such alternative design interpretations.

Let's return to our second example involving the schools and raters. If the same set is used in different schools, then raters are cross-classified with schools. An equation such as (18.1) can be used to model this situation, where now k refers to raters rather than neighbourhoods. If in addition, schools are crossed by neighbourhoods, then pupils are nested within a three-way rater / school / neighbourhood classification. For this case we may extend equation (18.1) by adding a term u_l for the rater classification as follows:

$$y_{i(jkl)} = \alpha + u_j + u_k + u_l + e_{i(jkl)} \tag{18.3}$$

If raters are not crossed with schools, but schools are crossed with neighbourhoods, a simple formulation might be:

$$y_{(ij)(kl)} = \alpha + u_k + u_l + e_{i(kl)} + e_{j(kl)} \tag{18.4}$$

where now i refers to pupils, j to raters, k to schools, and l to neighbourhoods.

Other applications are found, for example, in survey analysis where interviewers are crossed with enumeration areas.

18.2 How cross-classified models are implemented in MLwiN

Suppose we have a level 2 cross-classification with 100 schools drawing pupils from 30 neighbourhoods. If we sort the data into school order and ignore the cross-classification with neighbourhoods, the schools impose the usual block-diagonal structure on the $N \times N$ covariance matrix of responses, where N is the number of students in the data set. To incorporate a random neighbourhood effect we must estimate a non-block-diagonal covariance structure.

We can do this by declaring a third level in our model with one unit that spans the entire data set. We then create 30 dummy variables-one for each neighbourhood-and allow the coefficients of these to vary randomly at level 3 with a separate variance for each of our 30 neighbourhoods. We constrain all 30 variances to be equal.

We can allow other coefficients to vary randomly across schools by putting them in the model as level 2 random parameters in the usual way. If we wish the coefficient of a covariate — a slope — to vary randomly across neighbourhoods, the procedure is more complicated. We must create 30 new variables that are the product of the neighbourhood dummies and the covariate. These new variables are set to vary randomly at level 3. If we wish to allow intercept and slope to covary across neighbourhoods, we require 90 random parameters at level 3: an intercept variance, a slope variance and an intercept / slope covariance for each of the 30 neighbourhoods. As before we constrain the intercept variances, the covariances and the slope variances to produce three common estimates. The **SETX** command is provided to automate this procedure.

It is important to realise that although in this example we have set up a three-level MLwiN structure, conceptually we have only a two-level model, but with neighbourhood and school crossed at level 2. The third level is declared as a device to allow us to estimate the cross-classified structure. The details of this method are given in Rasbash & Goldstein (1994).

18.3 Some computational considerations

Cross-classified models can demand large amounts of storage for their estimation and can be computationally intensive. The storage required to run a model, in worksheet cells, is given by:

$$3n_e b + n_f b + \sum_{l=1}^{l=L} 4r_l b + 2br_{\max} \tag{18.5}$$

where

b is the number of level 1 units in the largest highest-level unit
n_e is the number of explanatory variables
n_f is the number of fixed parameters
L is the number of levels in the model
r_l is the number of variances being estimated at level l (covariances add no further space requirements)
$r_{\max}$ is the maximum number of variances at a single level

In cross-classified models, n_e will be large, typically the size of the smaller classification, and b will be equal to the size of the entire data set. These facts can lead to some quite devastating storage requirements for cross-classified models. In the example of 100 schools crossed with 30 neighbourhoods, suppose we have data on 3000 pupils. The storage required to handle the computation for a cross-classified variance components model is:

$$3n_eb + n_fb + \sum_{l=1}^{l=L} 4r_lb + 2br_{\max}$$

$$3 * 30 * 3000 + 3000 + 4(3000 + 3000 + 30 * 3000) + 2 * 3000 * 30$$

that is, some 840,000 free worksheet cells. If we wish to model a slope that varies across neighbourhoods, then the storage requirement doubles.

These storage requirements can be reduced if we can split the data into separate groups of schools and neighbourhoods. For example, if 40 of the 100 schools take children from only 12 of the 30 neighbourhoods, and no child from those 12 neighbourhoods goes to a different school, then we can treat those 40 schools by 12 neighbourhoods as a separate group. Suppose that in this way we can split our data set of 100 schools and 30 neighbourhoods into 3 separate groups, where the first group contains 40 schools and 12 neighbourhoods, the second contains 35 schools and 11 neighbourhoods and the third contains 25 schools and 7 neighbourhoods. We can then sort the data on school within group and make group the third level. We can do this because there is no link between the groups and all the covariances we wish to model are contained within the block diagonal matrix defined by the group blocks.

For a cross-classified variance components model n_e is now the maximum number of neighbourhoods in a group, that is 12, and b is the size of the largest group, say 1800. This leads to storage requirements of:

$$3 * 12 * 1800 + 1800 + 4(1800 + 1800 + 12 * 1800) + 2 * 1800 * 12$$

that is, about 210,000 free worksheet cells.

Finding such groupings not only decreases storage requirements but also significantly improves the speed of estimation. The command **XSEArch** is designed to find such groups in the data.

18.4 Modelling a two-way classification: An example

In this example we analyse children's overall exam attainment at age sixteen. The children are cross-classified by the secondary school and the primary school that they attended. The model is of the form given in equation (18.1) where u_j is a random departure due to secondary school and u_k is a random departure due to primary school. The data are from 3,435 children who attended 148 primary schools and 19 secondary schools in Fife, Scotland.

The initial analysis requires a worksheet size of just under 1000 k cells, less than the default size for an MLwiN worksheet. Note that the default size can be changed from 5000 k cells using the **Options** menu.

Retrieve the worksheet **xc.ws**. Opening the **Names** window shows that the worksheet contains 11 variables:

Variable	*Description*
VRQ	A verbal reasoning score resulting from tests pupils took when they entered secondary school.
ATTAIN	Attainment score of pupils at age sixteen.
PID	Primary school identifying code
SEX	Pupil's gender
SC	Pupil's social class
SID	Secondary school identifying code
FED	Father's education
CHOICE	Choice number of secondary school attended
MED	Mother's education
CONS	Constant vector
PUPIL	Pupil identifying code

In the following description we shall be using the **Command interface** window to set up and fit the models. A two-level variance components model with primary school at level 2 is already set up. To add the secondary school cross-classification, we need to do the following: create a level 3 unit spanning the entire data set; create the secondary school dummies; enter them in the model with random coefficients at level 3 and create a constraint matrix to pool the 19 separate estimates of the secondary school variances into one common estimate. We declare the third level by typing the following

command:

```
► IDEN 3 'CONS'
```

The remaining operations are all performed by the **SETX** command, whose syntax is given at the end of this chapter.

The first component specified on this command is the set of columns with random coefficient at the pseudo-level introduced to accommodate the cross-classification; in our case, this is a single column, **CONS**. The pseudo-level (3) is specified next. Then comes the column containing the identifying codes for the non-hierarchical classification, which in our case is **SID**.

The **SETX** command requires the identifying codes for the cross-classified categories to be consecutive and numbered from 1 upwards. If a different numbering scheme has been used, identifying codes can be put into this format using the **MLREcode** command. Here is an example:

```
► NAME C12 'NEWSID'
► MLRE 'CONS' 'SID' 'NEWSID'
```

Our secondary and primary identifying codes are already in the correct format so we do not need to use the **MLREcode** command.

The dummies for the non-hierarchical classification are written to the next set of columns. The dummies output are equal in number to k*r where k is the number of variables in <explanatory variable group> and r is the number of identifying codes in <ID column>. They are ordered by identifying code within explanatory variable. The constraint matrix is written to the last column. In addition the command sets up the appropriate random parameter matrix at level 3.

To set up our model, enter the following command:

```
► SETX 'CONS' 3 'SID' C101-C119 C20
```

If you examine the settings by typing the **SETT** command, you will see in the **Output** window that the appropriate structures have been created. Before running the model we must activate the constraints by typing

```
► RCON C20
```

The model will take some time to run. Four iterations are required to reach convergence. The results are as follows:

Parameter	Description	Estimate (SE)
σ^2_{uj}	Between-primary-school variance	1.12 (0.20)
σ^2_{uk}	Between-secondary-school variance	0.35 (0.16)
σ^2_e	Between-individual variance	8.1 (0.2)
α	Mean achievement	5.50 (0.17)

This analysis shows that the variation in achievement at age sixteen attributable to primary school is three times greater than the variation attributable to secondary school. This type of finding is an intriguing one for educational researchers and raises many further issues for further study.

Explanatory variables can be added to the model in the usual way to attempt to explain the variation.

We must be careful if we wish to create contextual secondary school variables using the ML** family of commands (or equivalent instructions via the **Multilevel Data Manipulation** window). The data are currently sorted by primary school, not secondary school, as these manipulations require. Therefore the data must be sorted by secondary school, the contextual variable created, and the data re-sorted by primary school.

18.5 Other aspects of the SETX command

When more than one coefficient is to be allowed to vary across a non-hierarchical classification, in some circumstances you may not wish the covariance between the coefficients to be estimated. This restriction can be achieved most easily by using two successive **SETX** commands. The following three examples illustrate this approach.

Example 1

```
► SETX 'CONS' 3 'SID' C101-C119 C20
```

This sets up the data for the cross-classified variance components model we have just run.

Example 2

Assuming the model is set up as in example 1 and the constraint matrix is activated, if we now type

```
► SETX 'VRQ' 3 'SID' C121-C139 C20
```

we shall have the structure for estimating the variance of the coefficients of **VRQ** and **CONS** across secondary schools. The intercept / slope covariance will not be estimated. If you want to run this model, you will be told that you need to increase the worksheet size.

Example 3

The commands shown in this example section 3 are for demonstration only. The model suggested here will not converge with the current data set.

If no cross-classified structure has yet been specified and we type

```
► SETX 'CONS' 'VRQ' 3 'SID' C101-C119 C121-C139 C20
```

we shall have the structure for estimating the variances of the coefficients of **VRQ** and **CONS** across secondary schools *and their covariance*. If you wish to issue this **SETX** command following the previous analysis you must first remove all the explanatory dummy variables (see below).

The **SETX** command adds the constraints it generates to any existing constraints that have been specified with the **RCON** command. Any additional random-parameter constraints must be activated using **RCON** before issuing any new **SETX** command. In particular, when elaborating cross-classified models with more than one **SETX** command, you must be sure to activate the constraint column generated by the first **SETX** before issuing second and subsequent **SETX** commands. Failure to do so will cause the first set of constraints not to be included in the constraints output by the second **SETX** command.

One limitation of the **SETX** command is that it will fail if any of the dummy variables it generates are already in the explanatory variable list. One situation where this may occur is when we have just estimated a cross-classified variance components model and we wish to expand it to a cross-classified random coefficient regression model in which slope / intercept covariances are to be estimated. In this case typing

```
► SETX 'CONS' 'VRQ' 3 'SID' C101-C119 C121-C139 C20
```

will produce an error message, since **C101-C119** will already be in the explanatory variable list. The problem can be avoided by removing **C101-C119** and then entering the **SETX** command:

```
► EXPL 0 C101-C119
► SETX 'CONS' 'VRQ' 3 'SID' C101-C119 C121-C139 C20
```

*Note that if the random constraint matrix is left active, the above **EXPL 0** command will remove from the matrix the constraints associated with **C101-C119**, leaving only those that the user had previously specified.*

After a **SETX** command, estimation must be restarted using the **STARt** command or button.

18.6 Reducing storage overhead by grouping

We can increase speed and reduce storage requirements by finding separate groups of secondary/primary schools as described above. The **XSEArch** command will do this.

Retrieve the original worksheet in xc.ws. We can search the data for separated groups by typing

```
► XSEArch 'PID' 'SID' C13 C14
```

Looking at **C13**, the column of separated groups produced, in the **Names** window, we see that it is a constant vector. That is, no separation can be made and all primary and secondary schools belong to one group. The new category codes in **C14** therefore span the entire range (1 to 19) of categories in the original non-hierarchical classification. This is not surprising since many of the cells in the 143 by 19 table contain very few individuals. It is this large number of almost empty cells that makes separation impossible. In many circumstances we may be prepared to sacrifice some information by omitting cells with very few students. We can omit data for cells with less than a given number of individuals using the **XOMIt** command.

In our case we can omit cells containing 2 or fewer members by typing

```
► XOMIt 2 C3 C6 C1-C2 C4-C5 C7-C11 C3 C6 C1-C2 C4-C5 C7-C11
```

If we now repeat the **XSEArch** command exactly as before, we find that **c13**, the group code column, contains 6 unique group codes (1, 2, 3, 5, 7 and 13) indicating that six groups have been found. The new category codes have a range from 1 to 8 indicating that the maximum number of secondary schools in any group is eight. You can use the **Tabulate** window to produce tables of secondary school and primary school by group. This confirms that with our reduced data set no primary school or secondary school crosses a group boundary.

We now sort the data by primary school within separated group. The group codes are now used to define a block diagonal structure for the variance-covariance matrix at level 3, which reduces the storage overhead and speeds up estimation. The following commands set up the model:

```
► SORT 2 c13 c3 C1 C2 C4-C11 C14 C13 C3 C1 C2 C4-C11 C14
► IDEN 3 C13
► SETX 'CONS' 3 C14 C101-C108 C20
► RCON C20
```

Notice that the new category codes in **C14** running from 1 to 8 (the maximum number of secondary schools in a separated group) are now used as the category codes for the non-hierarchical classification. This means we now need only 8 as opposed to 19 dummies to model this classification.

Estimation proceeds more than four times faster than in the full model, with very similar results.

Parameter	**Description**	**Estimate(se)**
σ^2_{uj}	between primary school variance	1.10(0.20)
σ^2_{uk}	between secondary school variance	0.38(0.19)
σ^2_e	between individual variance	8.1(0.2)
α	mean achievement	5.58(0.18)

18.7 Modelling a multi-way cross-classification

The commands described above can also be used to model multi-way classifications. For example, our secondary school by primary school cross-

classification could be further crossed, say by neighbourhoods, if neighbourhood identification was available.

In general we can model an n-way classification by repeated use of the **XSEArch** command to establish a separated group structure and then repeated use of the **SETX** command to specify each classification.

18.8 MLwiN commands for cross-classifications

The commands used here are described in the MLwiN **Help** system. Their syntax is as follows:

XOMIt

XOMIt cells with not more than <*value*> members from the cross-classification defined by <*input column-1*> and <*input column-2*> {carrying data in <*input data group*>} results to <*output column-1*> <*output column-2*> {and carried data to <*output data group*>}

XSEArch

XSEArch for separable groups in the cross-classification defined by <*column-1*> and <*column-2*> putting separated group codes in <*group ID column*> and new categories in <*new ID column*>

The first two columns describe the cross-classification to be searched. The non-hierarchical classification is specified by <*column-2*>. If separable groups can be found, they are assigned group codes 1, 2, etc. and these are placed in <*group ID column*>. The category codes of <*column-2*> are then recoded in <*new ID column*> to run from 1 within each group.

SETX

SETX set a random cross-classification, with coefficients of <*explanatory variable group*> random at level <*value*> across categories in <*ID column*>, storing dummies in <*output group*> and constraints in <*constraints column*>

<*explanatory variable* group> specifies the variables whose coefficients

we wish to vary randomly across the non-hierarchical classification.

Chapter learning outcomes

- ⋆ What a cross-classification is
- ⋆ How to set up and fit a cross-classified model

Chapter 19

Multiple Membership Models

Multiple membership models are used in situations where level 1 units belong to two or more higher-level units. In a longitudinal study of school students, for example, many will change their schools and thus 'belong' to more than one school during the study. When modelling such data, a student receives a weighted combination of residuals from all the schools to which the student belongs. To allocate the school effects appropriately, we need to construct a set of weights for each student that specify the student's school membership pattern.

19.1 A simple multiple membership model

Let's examine a simple variance components model of this kind. Suppose that we know, for each individual, the weight π_{ij_2}, associated with the j_2-th secondary school for student i (for example, the proportion of time spent in that school) with $\sum_{j_2=1}^{J_2} \pi_{ij_2} = 1$.

$$
\begin{aligned}
y_{i(j_2)} &= (X\beta)_{i(j_2)} + \sum_{j_2} u_{j_2}^{(2)} \pi_{ij_2} + e_{i(j_2)} \\
\sum_{j_2} u_{j_2}^{(2)} \pi_{ij_2} &= \pi_i u^{(2)} \\
u^{(2)^T} &= \{u_1^{(2)}, \ldots u_{J_2}^{(2)}\} \\
\pi &= \{\pi_1, \ldots \pi_{J_2} A\} \\
\pi_{j_2}^T &= \{\pi_{1j_2}, \ldots \pi_{Nj_2}\}
\end{aligned}
$$

where N is the total number of students and $u^{(2)}$is the $J_2 \times 1$ vector of

secondary school effects. This is therefore a two-level model in which the level 2 variation among secondary schools is modelled using the J_2 sets of weights for student i $(\pi_1, \ldots \pi_{J_2})$ as explanatory variables, with π_{j_2} the N x 1 vector of student weights for the j_2th secondary school. We have

$$\begin{aligned} \text{var}(u_{j_2}^{(2)}) &= \sigma_{u2}^2 \\ \text{cov}(u_{j_1}^{(1)} u_{j_2}^{(2)}) &= 0 \\ \text{var}(\sum_{j_2} u_{j_2}^{(2)} \pi_{ij_2}) &= \sigma_{u2}^2 \sum_{j_2} \pi_{ij_2}^2 \end{aligned}$$

These models can also be extended to deal with cases where higher-level unit identifications are missing. For details of these models with an example see Hill & Goldstein (1998).

There are two new commands that together can be used for specifying such multiple membership models. These are **WTCOl** and **ADDM**.

> *Note that the last letter of the command **WTCOl** is a letter 'l' rather than a number '1'.*

Let's first consider the use of the **WTCOl** command. Suppose we have a model with pupils nested within schools and we have only one response per pupil. However, some pupils attend more than one school during the study and we know the identities of the schools they attended and have information on how much time they spent in each school.

Similarly to a cross-classified model, we create a set of indicator variables, one for each school. Where a pupil attends more than one school they require the indicator variable for each school they attended to be multiplied by a weight, which for example could be based upon the proportion of time the pupil spent at that school. The indicator variables for all the schools the pupil did not attend are set to zero. It is this set of weighted indicator variables that is made to have random coefficients at level 2. As with cross-classified models, level 3 is set to be a unit spanning the entire data set and the variances of all the indicator variable coefficients are constrained to be equal.

The **WTCOl** command can be used to create the weighted indicator variables. If we have 100 schools and the maximum number of schools attended by a pupil is 4 then the **WTCOl** command would be

```
► WTCOl 4, id columns C1-C4, weights in C5-C8, weighted
  indicator columns output to C101-C200
```

Suppose pupil 1 attends only school 5, pupil 2 attends schools 8 and 9 with proportions 0.4 and 0.6 and pupil 3 attends schools 4, 5, 8 and 6 with proportions 0.2, 0.4, 0.3 and 0.1. Then the id and weight columns for these 3 pupils would contain the data

c1	c2	c3	c4	c5	c6	c7	c8
5	0	0	0	1	0	0	0
8	9	0	0	0.4	0.6	0	0
4	5	8	6	0.2	0.4	0.3	0.1

The first 9 columns of the output for these three children would be

c101	c102	c103	c104	c105	c106	c107	c108	c109
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0.4	0.6
0	0	0	0.2	0.4	0.1	0	0.3	0

The second command is the **ADDM** command. This command adds sets of weighted indicator variables created by the **WTCOl** command to the random part of the model at level M and generates a constraint matrix that defines the variances estimated for each set of indicators to be equal. It is possible to have more than one set of indicators if you wish to allow several random coefficients to vary across the multiple membership classification.

Continuing from the example outlined in the description of the **WTCOl** command, we first consider a variance components multiple membership model. In this case we enter:

```
► ADDM 1 set of indicators at level 2, in C101-C200,
  constraints to C20
```

If we wish to allow the slope of an X variable, say "PRETEST", in addition to the intercept, to vary randomly across the multiple membership classification, we must then form another set of indicators that are the product of the original indicators and "PRETEST". To do this enter:

```
► LINK C101-C200 G1
► LINK C201-C300 G2
► CALC G2=G1*'PRETEST'
► ADDM 2 sets of indicators at level 2, first set in
  C101-C200, second set in C201-C300, constraints to C20
```

The first **LINK** command puts the original indicators in group 1 (**G1**), and the second sets up a group (**G2**) for interactions. The **CALC** command creates the interactions. The **ADDM** command will place the two sets of indicators in the random part of the model as well as associated covariance terms between the two sets and establish the appropriate constraints in **C20**.

*Note that the **ADDM** command will not add its constraints to any existing constraints in the model.*

19.2 MLwiN commands for multiple membership models

The commands used here are described in the MLwiN **Help** system. Their syntax is as follows:

WTCOl <value> id columns <group 1> weights in columns <group 2> weighted indicator columns to <group 3>

ADDM <value> sets of indicators at level <value>, first set in <group>, second set in <group>, ..., constraints to <column>

Chapter learning outcomes

- ⋆ What a multiple membership model is
- ⋆ How to specify a multiple membership model in MLwiN

Bibliography

Aitkin, M. & Longford, N. (1986). Statistical modelling in school effectiveness studies (with discussion). *Journal of the Royal Statistical Society, Series A*, **149**:1–43.

Amin, S., Diamond, I. & Steele, F. (1997). Contraception and religiosity in Bangladesh. In G.W. Jones, J.C. Caldwell, R.M. Douglas & R.M. D'Souza, eds., *The continuing demographic transition*, pages 268–289. Oxford: Oxford University Press.

Atkinson, A.C. (1986). Masking unmasked. *Biometrika*, **73**:533–541.

Barnett, V. & Lewis, T. (1994). *Outliers in statistical data*. New York: John Wiley, 3rd edition.

Belsey, D.A., Kuh, E. & Welsch, R.E. (1980). *Regression diagnostics*. New York: John Wiley.

Browne, W.J. (1998). *Applying MCMC methods to multilevel models*. Ph.D. thesis, University of Bath.

Browne, W.J. (2003). *MCMC Estimation in MLwiN*. London: Institute of Education.

Bryk, A.S. & Raudenbush, S.W. (1992). *Hierarchical linear models*. Newbury Park, California: Sage.

Carpenter, J., Goldstein, H. & Rasbash, J. (2003). A novel bootstrap procedure for assessing the relationship between class size and achievement. *Journal of the Royal Statistical Society, Series C*, **52**:431–443.

Collett, D. (1991). *Modelling binary data*. New York: Chapman & Hall.

Darlington, R. (1997). Transforming a variable to a normal distribution or other specified shape. `http://comp9.psych.cornell.edu/Darlington/transfrm.htm`. Retrieved 21st June 2003.

Diggle, P. & Kenward, M.G. (1994). Informative dropout in longitudinal data analysis (with discussion). *Journal of the Royal Statistical Society, Series C*, **43**:49–93.

Efron, B. & Tibshirani, R. (1993). *An introduction to the bootstrap*. New York: Chapman & Hall.

Gelman, A., Roberts, G.O. & Gilks, W.R. (1995). Efficient Metropolis jumping rules. In J.M. Bernado, J.O. Berger, A.P. Dawid & A.F.M. Smith, eds., *Bayesian Statistics 5*, pages 599–607. Oxford: Oxford University Press.

Gilks, W.R., Richardson, S. & Spiegelhalter, D.J. (1996). *Markov Chain Monte Carlo in practice*. London: Chapman & Hall.

Goldstein, H. (1979). *The design and analysis of longitudinal studies*. London: Academic Press.

Goldstein, H. (1995). The multilevel analysis of growth data. In R. Hauspie, G. Lindgren & F. Falkner, eds., *Essays on Auxology*. Welwyn Garden City, England: Castlemead.

Goldstein, H. (1996). Consistent estimators for multilevel generalised linear models using an estimated bootstrap. *Multilevel Modelling Newsletter*, **8(1)**:3–6.

Goldstein, H. (2003). *Multilevel statistical models*. London: Arnold, 3rd edition.

Goldstein, H. & Healy, M.J.R. (1995). The graphical presentation of a collection of means. *Journal of the Royal Statistical Society, Series A*, **158**:175–7.

Goldstein, H. & Rasbash, J. (1996). Improved approximations for multilevel models with binary responses. *Journal of the Royal Statistical Society, Series A*, **159**:505–13.

Goldstein, H. & Speigelhalter, D.J. (1996). League tables and their limitations: statistical issues in comparisons of institutional performance. *Journal of the Royal Statistical Society, Series A*, **159**:385–443.

Goldstein, H., Rasbash, J., Yang, M., Woodhouse, G. et al. (1993). A multilevel analysis of school examination results. *Oxford Review of Education*, **19**:425–433.

Goldstein, H., Healy, M.J.R. & Rasbash, J. (1994). Multilevel time series models with applications to repeated measures. *Statistics in Medicine*, **13**:1643–55.

Goldstein, H., Rasbash, J. & Browne, W.J. (2002). Partitioning variation in multilevel models. *Understanding Statistics*, **1**:223–231.

Heath, A., Yang, M. & Goldstein, H. (1996). Multilevel analysis of the changing relationship between class and party in Britain 1964-1992. *Quality and Quantity*, **30**:389–404.

Hill, P.W. & Goldstein, H. (1998). Multilevel modelling of educational data with cross-classification and missing identification of units. *Journal of Educational and Behavioural statistics*, **23**:117–128.

Huq, N.M. & Cleland, J. (1990). Bangladesh fertility survery, 1989. Dhaka: National Institute of Population Research and Training (NIPORT).

Kendall, M. & Stewart, A. (1997). *The Advanced Theory of Statistics*, volume 1. New York: Macmillan Publishing, 4th edition.

Langford, I.H. & Lewis, T. (1998). Outliers in multilevel models (with discussion). *Journal of the Royal Statistical Society, Series A*, **161**:121–160.

Langford, I.H., Bentham, G. & McDonald, A. (1998). Multilevel modelling of geographically aggregated health data: A case study on malignant melanoma mortality and UV exposure in the European Community. *Statistics in Medicine*, **17**:41–58.

Lawrence, A.J. (1995). Deletion, influence and masking in regression. *Journal of the Royal Statistical Society, Series B*, **57**:181–189.

Longford, N.T. (1993). *Random coefficient models*. Oxford: Clarendon Press.

McCullagh, P. & Nelder, J. (1989). *Generalised linear models*. London: Chapman & Hall.

Nuttall, D.L., Goldstein, H., Prosser, R. & Rasbash, J. (1989). Differential school effectiveness. *International Journal of Educational Research*, **13**:769–776.

Plewis, I. (1997). *Statistics in education*. London: Arnold.

Raftery, A.E. & Lewis, S.M. (1992). How many iterations in the Gibbs sampler? In J.M. Bernado, J.O. Berfer, A.P. Dawid & A.F.M. Smith, eds., *Bayesian Statistics 4*, pages 765–76. Oxford: Oxford University Press.

Rasbash, J. & Goldstein, H. (1994). Efficient analysis of mixed hierarchical and cross-classified random structures using a multilevel model. *Journal of Educational and Behavioural Statistics*, **19**:337–50.

Rowe, K.J. (2003). Estimating interdependent effects among multilevel composite variables in psychosocial research: An example of the application of multilevel structural equation modeling. In S.P. Reise & N. Duan, eds., *Multilevel modeling: Methodological advances, issues, and applications*, pages 255–284. Mahwa, NJ: Erlbaum Associates.

Rowe, K.J. & Hill, P.W. (1998). Modeling educational effectiveness in classrooms: The use of multilevel structural equations to model students' progress. *Educational Research and Evaluation*, **4(4)**:307–347.

Silverman, B.W. (1986). *Density estimation for statistics and data analysis*. London: Chapman & Hall.

Snijders, T.A.B. & Bosker, R.J. (1999). *Multilevel Analysis*. Newbury Park, California: Sage.

Spiegelhalter, D.J., Best, N.G., Carlin, B.P. & van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B*, **64**:191–232.

Tizard, B., Blatchford, P., Burke, J. & Farquhar, C. (1988). *Young children at school in the inner city*. Hove, Sussex: Lawrence Erlbaum.

Velleman, P.F. & Welsch, R.E. (1981). Efficient computing of regression. *American Statistician*, **35**:234–242.

Venables, W.N. & Ripley, B.D. (1994). *Modern applied statistics with S-Plus*. New York: Springer-Verlag.

Woodhouse, G. & Goldstein, H. (1989). Educational performance indicators and LEA league tables. *Oxford Review of Education*, **14**:301–319.

Yang, M. & Woodhouse, G. (2001). Progress from GCSE to A and AS level: Simple measures and complex relationships. *British Educational Research Journal*, **27**:245–268.

Yang, M., Goldstein, H., Rath, T. & Hill, N. (1999). The use of assessment data for school improvement purposes. *Oxford Review of Education*, **25**:469–483.

Index

ADDM command, 288–290
Analysis of Variance, 9, 15, 22, 25–29, 33, 34, 37, 40, 60, 196
Antilogit transformation, 173, 181, 183

Bangladesh Fertility Survey, 119
Basic Statistics menu, 13, 14, 21, 25, 26, 84, 122, 127, 148, 150, 226, 246, 248, 250, 254
Bayesian modelling, xvi
Binary response model, 119, 121, 123–125, 129, 134, 141, 145, 148, 152, 153, 156, 157, 194, 225, 226
Binomial error, 119, 124, 125, 141, 145, 185, 194, 226, 254
Binomial response model, 119, 124, 125, 141, 145, 185, 194, 226, 254
Block diagonal structure, 277
Bootstrapping, 133, 194, 245, 248, 249, 251–253, 257, 259–261, 263–273
Boxes in worksheet, xiv, xv, 41, 174, 265
British Election Study, 261, 270
BUGS software, xviii

Calculate command, 252, 290
Case-sensitive, xv
Categorical explanatory variables, 17, 18, 23, 110, 114, 124, 136, 152, 169, 186, 216
Categorical response variable, 119, 121, 147, 156, 157, 163, 164, 169, 213, 226
Category names, 113
Caterpillar plot, 43, 68, 69, 73–75, 223, 241
Causality, 3
Centre for Multilevel Modelling, xiii, xviii
Chemistry A-level data, 163, 167, 170, 173, 178, 183
Clustered data, xiv, 1–3, 28, 30, 32, 58, 107, 114, 116, 119, 130, 156, 188, 195, 200, 216, 230, 275, 280, 281, 283, 284, 286
Command interface, xiv, xvi, 93, 135, 142, 165, 166, 181, 187, 209, 249, 265, 280
Complementary log-log link function, 121, 124
Complex level 1 variation, 102, 108, 208
Computational considerations, 277
Confidence interval, 14, 33, 43, 160, 233, 248, 251, 253, 259, 261, 268
Confounding in ANOVA, 27, 29, 142
Contextual effects, 85–87, 281
Contraceptive use, 119, 122, 126, 129, 130, 136, 137, 141, 147, 150, 155, 156
Convergence, 32, 75, 108, 118, 130, 132, 157, 205, 206, 208, 209, 260, 261, 265, 266, 281
Copying data from other applications, 111
Count data, 185–188, 190, 194, 226
Coursework examination, 213–215, 217, 220–224
Covariance matrix, 61, 105, 106, 131, 153, 158, 168, 169, 174, 216, 218, 219, 221, 269, 270, 277, 284
Cross classification, 275–282, 285, 288
Cumulative response probabilities, 168, 182

Data exploration, 203, 229
Data input, xiv, 109–112

Data Manipulation menu, 12, 55, 93, 112, 115, 124, 134, 135, 142–144, 150, 155, 160, 165, 200, 204, 217, 225, 254, 281
Data output, xiv, 109–112
Data window, 12, 71, 72, 78, 152, 217, 231
Deletion residuals, 235
Demography, 119, 122, 126, 129, 130, 136, 137, 141, 147, 148, 150, 154–156, 160, 161
Denominator in logistic regression, 124, 141–143, 145, 153, 171, 226
Dependent variable, 2, 4, 8, 16, 33, 39, 43, 92, 117, 119, 124, 141–143, 147, 148, 152, 163–165, 168, 170, 171, 176, 185, 198, 214, 216, 226, 230–232, 261, 262, 269
Deviance, 20, 62, 84, 86, 206, 222, 240–242
DFITS, 237
Diagnostic information, xiv, xvi, 41, 194, 229, 235–237
Diagnostics, xiv, xvi, 41, 194, 229, 235–237
Discrete response data, xvi, xvii, 119, 121, 123, 129, 130, 133, 134, 141, 145, 148, 152, 153, 156, 157, 169, 185–188, 190, 193, 194, 224–226, 245, 259
Display precision, 199, 209

Educational achievement, 2, 47, 51, 59, 82, 107, 108, 167, 196, 275, 281, 284
Email discussion group, xviii
Environmental health, 185, 186, 188, 190, 193
Equations window, xvii, 15–19, 23, 26, 30, 33, 38, 39, 48–50, 58, 61, 75, 82, 84, 85, 92–99, 102, 104–106, 114, 124, 125, 128–130, 132, 137, 142, 145, 151–154, 157, 165, 169–174, 187, 188, 193, 204, 205, 208, 216–218, 226, 231, 241
ESRC, xviii
Estimate tables window, 159, 220
Estimation control, 32, 105, 263, 265, 271, 272
Estimation menu, 32, 105, 189, 263, 265, 271, 272
Examination data, 2–6, 10, 13, 14, 20, 35, 37, 164, 214
EXCEL, 111
Explanatory variable, 2, 7, 8, 10, 15, 27, 29, 52, 59, 60, 70, 83, 93, 97, 99–102, 107, 115, 118, 121, 123, 124, 128, 129, 131, 133–138, 140, 142, 145, 148–150, 152, 153, 155, 161, 165, 166, 170–172, 176, 177, 182, 186, 197, 205, 214, 217, 229, 238, 262, 275, 278, 280, 282, 283, 286, 288

F-test, 25, 26
Fixed effects model, 9, 15, 22, 25–29, 33, 34, 37, 40, 60, 196
Fixed parameter, 7, 8, 37, 43, 52, 206, 209, 231, 240, 259, 260, 269, 278
Fixed part of a multilevel model, 6, 8, 19, 33, 51, 61, 70, 175, 178, 181, 190, 209, 210, 219, 239, 240

GCSE, 163, 164, 166, 167, 178–183, 213
GCSE score, 177
General notation, 15, 22, 92, 93, 97, 99
Generalised linear multilevel model, xvi, 119, 121, 123, 129, 134, 141, 145, 148, 152, 153, 156, 157, 194, 225, 226
Graph window, 42, 47, 51–53, 55–57, 63, 64, 67–73, 75–78, 88, 100, 103, 106, 107, 164, 192, 193, 211, 224, 232, 234, 235, 237, 238, 240, 242, 243, 247, 251, 255–257, 264, 266–268, 272
Graphical user interface, xiv
Growth curve, 196, 197, 205

Help keywords, xvi

Help system, xvi, 12, 13, 220, 250
Hierarchical data, 1–3, 28, 30, 32, 58, 114, 119, 130, 156, 188, 200, 230, 275, 280, 281, 283, 284, 286
Hierarchy Viewer, 32
Household, 1, 2, 39
Hypertext links, xvi
Hypothesis test, 127

Inferences, 5, 6, 9, 20, 27, 41, 58, 60, 180, 248, 259
Influence, 229, 230, 235, 237, 238
Input and output of data, xiv, 109–112
Installing MLwiN, xiii
Interaction effect, 60, 84–87
Intercept, 5–7, 10, 15, 20, 22, 29, 30, 33, 47, 49–51, 54–56, 60–62, 64, 67, 73, 75–78, 84, 86, 87, 91, 93, 95, 97, 99, 100, 118, 129, 130, 132, 133, 145, 149, 165, 171, 172, 174, 176, 181, 190, 193, 215, 217, 221, 223, 231, 233–237, 240–242, 270, 277, 282, 289
Intervals and tests window, 127, 129, 133, 139
Intra-class correlation, 29, 34, 104, 119, 133–136
Iterated bootstrap, 260
Iterative generalised least squares (IGLS), 26, 32, 33, 118, 130, 188, 189, 215, 245, 260, 262, 263, 266, 271

Joint Information Systems Committee, xviii

Kernel density estimate, 251, 253, 268, 273

Level 2 variable, 27, 34, 140, 141
Leverage, 235, 237
Likelihood ratio test, 9, 14, 26, 33, 34, 51, 114, 133
Link function, 119, 121, 124, 128, 134, 148, 151, 168, 169, 186
Log-likelihood value, 20, 62, 84, 86, 206, 222, 240–242
Log-log link function, 121, 124
Logistic regression, 119, 121, 122, 125, 128, 134, 141, 147–149, 183
Logit link function, 119, 121, 122, 124, 125, 128, 145, 148, 149, 151, 168–170, 183, 226
London Reading Test (LRT), 2–6, 10

Macro, xvii, 133–135, 155, 156, 196, 212, 225, 249–252, 254
Macro editor, 249
Malignant melanoma, 185, 186, 188, 190, 193
Markov Chain Monte Carlo (MCMC), 118, 130, 133, 157, 159, 176, 194, 245, 250, 251, 253, 259, 261
Maximum likelihood, 130, 196, 215
Measurement occasion, 1, 196–198
Missing data, 11, 111, 113, 196
MLn, xiv, xvi, xviii
Model checking and diagnosis, 40
MQL, 130–132, 145, 157–159, 175, 176, 188–190, 245, 262, 265, 271
Multilevel data, xiv, 1–3, 28, 30, 32, 58, 107, 114, 116, 119, 130, 156, 188, 195, 200, 216, 230, 275, 280, 281, 283, 284, 286
Multinomial response model (ordered), 163, 168, 169, 172, 173
Multinomial response model (unordered), 147–150, 156, 159, 170
Multiple membership model, xvii, 287–290
Multiple regression, xvii, 3, 4, 6, 9, 10, 15, 18, 20–22, 26, 27, 39, 48, 49, 51, 56, 58–61, 70, 77, 121, 122, 148, 150, 156, 229, 231, 233, 237, 243, 282
Multivariate model, 213, 214
Multivariate window, 212

Naming categories, 23, 113, 150, 169
Naming variables, xv, 11, 23, 112, 113, 120, 124, 144, 150, 169, 170, 174, 185, 199, 202, 204, 209, 214, 216, 254, 255, 261, 279, 283
NANOSTAT, xviii

Negative variance estimates, 105, 263, 265, 271
Neighbourhood, 275–278, 285
Nested models, 26
Nonlinear model estimation options, 101, 125, 132, 145, 154, 158, 172, 176, 188, 189
Normal distribution, 7, 10, 14, 19, 28, 31, 33, 43, 61, 92, 121, 126, 129, 131, 133, 134, 138, 154, 156, 188, 194, 215, 226, 246–248, 250–254
Normal distribution test, 9, 14, 15, 20, 25, 33, 34
Normal probability plot, 43, 44
Normal score, 10, 43, 165, 166, 177, 180, 207, 230, 238
Notation, xvii, 6–8, 15, 91, 92, 137, 152, 168
Null hypothesis, 14, 20, 25, 26, 62, 84, 127

Odds ratio, 119, 121, 122, 126, 149, 154, 155, 183
Offset term in Poisson models, 186, 187
Omitting cells of a cross classification, 283
Options menu, 33, 111, 113, 199, 209, 279
Ordered categorical response model, 163, 168, 169, 172, 173
Ordinary least squares (OLS), 6, 9, 15, 33, 39, 59, 229
Outcome variable, 2, 4, 8, 16, 33, 39, 43, 92, 117, 119, 124, 141–143, 147, 148, 152, 163–165, 168, 170, 171, 176, 185, 198, 214, 216, 226, 230–232, 261, 262, 269
Output window, xvi

P-value, 14, 25, 26, 127
Parameter estimates, xv, 15, 19, 20, 24, 30, 33, 48–50, 82, 83, 94, 187, 218, 230, 231, 240, 241, 259, 260
Parametric bootstrap, 249, 252, 254, 260, 269, 271–273
Poisson model for count data, 185–188, 190, 194, 226
Population, 1–3, 5–7, 14, 15, 20, 24, 26–29, 40, 60, 62, 153, 186, 246, 248, 249, 251, 252, 254
Posterior estimates, 5
PQL, 130–132, 145, 157–159, 176, 189, 190, 245, 265
Predicted probability, 121, 122, 126, 149, 155, 156
Predictions window, 51, 70, 71, 75, 77, 191, 192, 210, 242
Predictor variable, 2, 7, 8, 10, 15, 27, 29, 52, 59, 60, 70, 83, 93, 97, 99–102, 107, 115, 118, 121, 123, 124, 128, 129, 131, 133–138, 140, 142, 145, 148–150, 152, 153, 155, 161, 165, 166, 170–172, 176, 177, 182, 186, 197, 205, 214, 217, 229, 238, 262, 275, 278, 280, 282, 283, 286, 288
Prior distribution, 167
Probit link function, 119, 121, 124, 128, 134
Proportional odds model for ordered responses, 163, 168, 169, 172, 173
Proportions, 119, 124, 125, 141, 145, 185, 194, 226, 254
Pupils, 2, 3, 6, 7, 50, 82–84, 86, 87, 97, 164–166, 168–171, 173, 230, 238–240, 242, 243, 276, 288, 289

Quasi-likelihood, xvii, 130–133, 145, 157–159, 175, 176, 188–190, 245, 259, 262, 265, 271

Random classification, 5
Random coefficient model, 8, 59, 63, 67, 75, 97, 99, 100, 118, 119, 133, 134, 137, 138, 176, 204, 210, 231, 244, 270, 280, 282, 288, 289
Random intercept model, 51, 63, 67, 68, 97, 119, 129–131, 133, 134, 141, 145, 156, 157, 216

Random parameter, xiv, 7, 100, 102, 105, 206, 209, 220, 231, 241, 259, 269, 277, 280
Random part of a multilevel model, 7, 19, 30, 159, 231, 240, 241, 243, 289, 290
Random sample, 3, 5–7, 29, 60, 246
Random slope model, 8, 59, 63, 67, 75, 97, 99, 100, 118, 119, 133, 134, 137, 138, 176, 204, 210, 231, 244, 270, 280, 282, 288, 289
RCON command, 281, 282, 284
Reading attainment, 195, 197–204, 208
Reference category, 18, 23, 28, 35, 60, 83, 126, 148, 149, 152, 153, 168, 178, 191
Regression, xvii, 3, 4, 6, 9, 10, 15, 18, 20–22, 26, 27, 39, 48, 49, 51, 56, 58–61, 70, 77, 121, 122, 148, 150, 156, 229, 231, 233, 237, 243, 282
Repeated measurements, 195–198, 208, 212, 276
Residual, 6, 7, 15, 26, 28, 29, 39, 41, 42, 54–56, 73, 74, 133, 135, 136, 138, 140, 141, 160, 161, 210, 215, 223, 231, 233, 234, 237, 239–242
Residual +/- 1.96 sd x rank, 160
Residual plot, 43, 68, 69, 73–75, 223, 241
Residual variance, 20
Residuals window, 76, 224
Response variable, 4, 8, 16, 33, 39, 43, 92, 117, 119, 124, 141–143, 147, 148, 152, 163–165, 168, 170, 171, 176, 185, 198, 214, 216, 226, 230–232, 261, 262, 269
Retrieving data, 271
Retrieving files, xiv, 11, 35, 38, 44, 64, 82, 109, 114, 120, 135, 155, 225, 246, 249
RIGLS, 26, 118, 130, 188, 189, 245, 260, 262, 263, 266, 271
Saving data, 35, 44, 64, 68, 111, 114, 202, 203, 231, 263
School effectiveness, 5, 10, 25, 28
Scoring system, 163, 164, 230
Seed, 135, 265, 271
Serial correlation, 196, 212
SETX command, 277, 280–286
Shrinkage factor, 39, 40
Shrunken residual, 39
Simulations, 133, 134, 190, 245, 248, 249, 253, 260, 261, 266
Sorting data, 32, 114–116, 281
Spreadsheet, xiv, 11, 12
SPSS, 111
Standardised residual, 41, 43, 207, 237, 238
Storage requirements, 111, 112, 277–279, 282–284
Student progress, 3, 11, 33, 107, 167, 183, 264

T-test, 9, 14, 15, 20, 25, 33, 34
Threshold model for binary response data, 88, 134, 136
Tolerance, 33
Toolbar, xvi, 38, 49, 50, 58, 75, 83, 84, 112, 174, 216, 226, 263, 271
Trace (MCMC), 251
Trajectories window, 264–266, 268, 272, 273
Troubleshooting, 110

Units at a level, xiv, 1, 5, 17, 20, 28, 40, 43, 49, 60, 84, 114, 116, 118, 122, 124, 125, 130, 133, 142, 170, 186, 188, 190, 194–196, 200, 213, 214, 230, 237, 240, 244, 259, 268, 270, 278, 287
Unordered categorical response model, 147–150, 156, 159, 170

Variance components, 7, 28, 39, 51, 60, 118, 129, 188, 204, 260, 262, 278, 280–282, 287, 289
Variance function window, 91, 97, 99, 100, 102–107, 138, 208, 210

Variance partition coefficient, 29, 34, 104, 119, 133–136
Viewing data, 12
Voting behaviour, 2, 261–263

Wald test, 127, 129, 133, 139
Websites, xviii
Worksheet, xiv, 10–12, 21, 35, 44, 47, 54, 64, 67, 68, 97, 111, 112, 114, 120, 135, 199, 202, 203, 214, 225, 231, 246, 261, 263, 271, 277–279, 282, 283
WTCOl command, 288–290

XOMIt command, 283, 284, 286
XSEArch command, 279, 283–286